

ADAPTIVE SMOLYAK PSEUDOSPECTRAL APPROXIMATIONS

PATRICK R. CONRAD AND YOUSSEF M. MARZOUK

Abstract. Polynomial approximations of computationally intensive models are central to uncertainty quantification. This paper describes an adaptive method for non-intrusive pseudospectral approximation, based on Smolyak’s algorithm with generalized sparse grids. We rigorously develop and generalize the non-adaptive method proposed in [6], and compare it to a common alternative approach for using sparse grids to construct polynomial approximations, direct quadrature. Analysis of direct quadrature shows that $\mathcal{O}(1)$ errors are an intrinsic property of some configurations of the method, as a consequence of internal aliasing. We provide precise conditions, based on the chosen polynomial basis and quadrature rules, under which this aliasing error occurs. We then establish theoretical results on the accuracy of Smolyak pseudospectral approximation, and show that the Smolyak approximation avoids internal aliasing and makes far more effective use of sparse function evaluations. These results are applicable to broad choices of quadrature rule and generalized sparse grids. Exploiting this flexibility, we introduce a greedy heuristic for adaptive refinement of the pseudospectral approximation. We numerically demonstrate convergence of the algorithm on the Genz test functions, and illustrate the accuracy and efficiency of the adaptive approach on a realistic chemical kinetics problem.

Key words. Smolyak algorithms, sparse grids, orthogonal polynomials, pseudospectral approximation, approximation theory, uncertainty quantification

AMS subject classifications. 41A10, 41A63, 47A80, 65D15, 65D32

1. Introduction. A central issue in the field of uncertainty quantification is the response of a model to random inputs. When model evaluations are computationally intensive, techniques for *approximating* the model response in an efficient manner are essential. Approximations may be used to evaluate moments or the probability distribution of a model’s outputs, or to evaluate sensitivities of model outputs with respect to the inputs [20, 41, 33]. Approximations may also be viewed as *surrogate models* to be used in optimization [21] or inference [22], replacing the full model entirely.

Often one is faced with black box models that can only be evaluated at designated input points. We will focus on constructing multivariate polynomial approximations of the input-output relationship generated by such a model; these approximations offer fast convergence for smooth functions and are widely used. One common strategy for constructing a polynomial approximation is interpolation, where interpolants are conveniently represented in Lagrange form [1, 42]. Another strategy is projection, particularly orthogonal projection with respect to some inner product. The results of such a projection are conveniently represented with the corresponding family of orthogonal polynomials [4, 20, 43]. When the inner product is chosen according to the input probability measure, this construction is known as the (finite dimensional) polynomial chaos expansion (PCE) [14, 32, 8]. Interpolation and projection are closely linked, particularly when projection is computed via discrete model evaluations. Moreover, one can always realize a change of basis [9] for the polynomial resulting from either operation. Here we will favor orthogonal polynomial representations, as they are easy to manipulate and their coefficients have a useful interpretation in probabilistic settings.

This paper discusses *adaptive Smolyak pseudospectral approximation*, an accurate and computationally efficient approach to constructing multivariate polynomial chaos expansions. Pseudospectral methods allow the construction of polynomial approximations from point evaluations of a function [4, 3]. We combine these methods with *Smolyak’s algorithm*, a general strategy for sparse approximation of linear operators on tensor product spaces, which saves computational effort by weakening the assumed coupling between the input dimensions. Gerstner and Griebel [13] developed an adaptive variant of Smolyak’s algorithm for numerical integration and illustrated the effectiveness of on-the-fly heuristic adaptation. Here we extend their approach to the pseudospectral approximation of functions. Adaptivity is expected to yield substantial efficiency gains in high dimensions—particularly for functions with anisotropic dependence on input parameters and functions whose inputs might not be strongly coupled at high order.

Previous attempts to extend pseudospectral methods to multivariate polynomial approximation with sparse model evaluations employed ad hoc approaches that are not always accurate. A common procedure has been to use sparse quadrature, or even dimension-adaptive sparse quadrature, to evaluate polynomial coefficients directly [39, 20]. This leads to at least two difficulties. First, the truncation of the polynomial expansion must be specified independently of the quadrature grid, yet it is unclear how to do this, particularly for anisotropic and generalized sparse grids. Second, unless one uses excessively high-order quadrature, significant aliasing errors may result. Constantine *et al.* provided the first clear demonstration of these aliasing errors, and proposed a Smolyak algorithm that does not share them [6].

Our treatment places this solution in the broader context of Smolyak constructions, and explains the origin of the observed aliasing errors for general (e.g., anisotropic) choices of sparse grid and quadrature rule. The main contributions of this paper are twofold. The first is theoretical: we provide an error analysis that shows why the conventional approach (termed ‘direct quadrature’) goes wrong, and we develop polynomial exactness results for Smolyak pseudospectral approximation on *generalized* sparse grids. We do so without relying on properties of any particular quadrature rule, beyond univariate polynomial exactness. These results then provide a rigorous foundation for *adaptivity*: we introduce and demonstrate a fully adaptive algorithm for Smolyak pseudospectral approximation, which uses a single tolerance parameter to drive iterative refinement of both the polynomial approximation space and the collection of model evaluation points.

The remainder of this introduction provides a brief motivating example. Section 2 defines the essential building-block problems of integration and polynomial approximation. Section 3 develops *computable* one-dimensional and tensorized approximations for these problems, i.e., quadrature and pseudospectral approximation. Section 4 describes general Smolyak algorithms and their properties, then applies these results to quadrature and pseudospectral approximation, yielding our principal theorems about exactness of the Smolyak approximation. Section 5 compares the Smolyak approach to conventional direct quadrature. We show that, in almost all cases, direct quadrature is not an appropriate method for constructing polynomial expansions and should be superseded by Smolyak pseudospectral methods. In Section 6 we present an adaptive extension to the Smolyak approach. Sections 7 and 8 contain observations on the choice of quadrature rule and connections with polynomial interpolation. As the adaptive methods are heuristic, Section 9 provides numerical demonstrations of their usefulness.

1.1. Overview. Before a more formal exposition, we give a brief overview of the Smolyak approach and the results of our analysis. This section is intentionally loose with notation, preferring to focus on intuition.

A polynomial chaos expansion is a series expansion:

$$f(x) \approx \sum_i f_i \Psi_i(x) \quad (1.1)$$

in some basis of orthonormal polynomials Ψ_i [38, 43]. If the function f is square integrable, the series converges to f in mean square. Practical uses of polynomial chaos select some truncation of the series. Taking the inner product of both sides with a basis function yields an expression for the coefficients

$$\langle f, \Psi_j \rangle = \sum_i f_i \langle \Psi_i, \Psi_j \rangle = \sum_i f_i \delta_{ij} = f_j \quad (1.2)$$

where δ_{ij} is the Kronecker delta. The pseudospectral approach approximates this expression by substituting a quadrature operator \mathcal{Q} for the inner product [4]:

$$\tilde{f}_j = \int f(x) \Psi_j(x) dx \approx \mathcal{Q}(f \Psi_j) \quad (1.3)$$

This approach incurs aliasing errors because, while the basis functions are analytically orthogonal, some pairs of basis functions will not appear orthogonal when the inner product is approximated via quadrature. Specifically, if for some i, j , $\mathcal{Q}(\Psi_i \Psi_j) \neq \delta_{ij}$, we say that Ψ_i and Ψ_j are not *numerically orthogonal*. These quadrature inaccuracies cause aliasing, because one coefficient can “bleed” onto another.

EXAMPLE 1.1. Assume that $f(x) = \Psi_2(x) + 10^{-3}\Psi_{20}(x)$, so that $f_2 = 1$ and $f_{20} = 10^{-3}$. For the sake of this example, define three quadrature inner products as:

$$\mathcal{Q}(\Psi_2 \Psi_2) = 1 \tag{1.4}$$

$$\mathcal{Q}(\Psi_{20} \Psi_{20}) = 1 \tag{1.5}$$

$$\mathcal{Q}(\Psi_2 \Psi_{20}) = 0.1 \tag{1.6}$$

Hence, the quadrature computes the mixed inner product incorrectly. If we compute the coefficients with this quadrature rule, we find

$$\tilde{f}_2 = 1 * 1 + 0.01 * 10^{-3} = 1.0001 \tag{1.7}$$

$$\tilde{f}_{20} = 10^{-3} * 1 + 0.1 * 1 = .101 \tag{1.8}$$

Although \tilde{f}_2 is essentially correct, \tilde{f}_{20} is not even the correct order of magnitude; hence this approximation is likely unacceptable.

This example illustrates that it is crucial for pseudospectral methods to use quadrature that respects the orthogonality of the basis. In general, aliasing is an unavoidable consequence of the numerical approximation to the integral, so our goal is to limit its impact on the overall approximation. In one dimension, controlling the error is simple, but in higher dimensions the problem is more subtle. The naïve approach, direct quadrature, simply suggests using a sparse quadrature rule in (1.3). This approach can lose numerical orthogonality of the output basis, with disastrous results. This is precisely the source of error encountered by Constantine *et al.* [6].

The Smolyak algorithm resolves the potential for large errors by blending different full tensor approximations, which themselves are rigorously constructed to avoid certain aliasing errors; thus the Smolyak approximation inherits their favorable properties. The result is an algorithm that uses a sparse set of model evaluations, yet avoids the aliasing issues incurred by sparse direct quadrature schemes.

2. One Dimensional and Tensor Problems. We begin by presenting problems of integration and of approximation with polynomial chaos expansions. Exploiting the structure of tensor product problems is a core theme of this paper, so we will start in each case with a one-dimensional problem and then explain how to construct the corresponding multi-dimensional problem through tensorization. This section focuses on defining the problems and setting notation, leaving numerical approximations to the next section.

2.1. General setting. Consider a collection of one-dimensional linear operators $\mathcal{L}^{(i)}$, where (i) indexes the operators used in different dimensions.¹ We can extend a collection of these operators into higher dimensions by constructing the tensor product operator

$$\mathcal{L}^{(d)} := \mathcal{L}^{(1)} \otimes \dots \otimes \mathcal{L}^{(d)}. \tag{2.1}$$

The one-dimensional operators need not be identical; the properties of the resulting tensor operator are constructed independently from each dimension. The bold parenthetical superscript refers to the tensor operator instead of the constituent one-dimensional operators.

¹By ‘one-dimensional’ we mean linear operators on functions of one variable, i.e., $\mathcal{L}^{(i)}$ acts on functions of $x^{(i)}$. The tensor product operator in (2.1) then acts on functions of d variables, i.e., functions of $\mathbf{x} = (x^{(1)}, \dots, x^{(d)})$.

2.2. One-dimensional integration. Let $X^{(i)}$ be an open or closed interval of the real line \mathbb{R} . Then we define the weighted integral operator in one dimension as follows:

$$\mathcal{I}^{(i)}(f) := \int_{X^{(i)}} f(x) w^{(i)}(x) dx \quad (2.2)$$

where $f : X^{(i)} \rightarrow \mathbb{R}$ is some real-valued function and $w^{(i)} : X^{(i)} \rightarrow \mathbb{R}^+$ is an integrable weight function.

2.3. Multi-dimensional integration. One-dimensional integration extends easily to higher-dimensional integrals over product domains with separable weight functions. Let $X^{(1)}, \dots, X^{(d)}$ be a collection of intervals on the real line, as in the one dimensional case. Then let

$$\mathbf{X} := X^{(1)} \times \dots \times X^{(d)} \subseteq \mathbb{R}^d \quad (2.3)$$

be the domain of integration, defined by the Cartesian product of one-dimensional intervals. Consider a real-valued function defined on this product space, $f : \mathbf{X} \rightarrow \mathbb{R}$. Then the multi-dimensional integral is given by the tensor product of one-dimensional integral operators:

$$\begin{aligned} \mathcal{I}^{(\mathbf{d})}(f) &= \mathcal{I}^{(1)} \otimes \dots \otimes \mathcal{I}^{(d)}(f) \\ &= \int_{X^{(1)}} \dots \int_{X^{(d)}} w^{(1)}(x^{(1)}) \dots w^{(d)}(x^{(d)}) f(x^{(1)}, \dots, x^{(d)}) dx^{(d)} \dots dx^{(1)} \\ &= \int_{\mathbf{X}} w(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (2.4)$$

$$\text{where } w(\mathbf{x}) := \prod_{i=1}^d w^{(i)}(x^{(i)}).$$

2.4. One-dimensional polynomial chaos expansions. A polynomial chaos expansion approximates a function with a weighted sum of orthonormal polynomials [38, 43]. Let $\mathcal{H}^{(i)} := \{f : X^{(i)} \rightarrow \mathbb{R}\}$ be a separable Hilbert space of square-integrable functions, with inner product for $f, g \in \mathcal{H}^{(i)}$ defined as:

$$\langle f, g \rangle := \int_{X^{(i)}} f(x) g(x) w^{(i)}(x) dx = \mathcal{I}^{(i)}(fg). \quad (2.5)$$

Further, let the weight function be normalized such that $\langle 1, 1 \rangle = 1$, so that $w^{(i)}(x)$ may represent a probability density. There exist a set of polynomials $\{\psi_j^{(i)}(x) : j \in \mathbb{N}_0\}$ orthonormal with respect to this weight. Here \mathbb{N}_0 is the set of natural numbers including zero, and orthonormality requires $\langle \psi_j^{(i)}(x), \psi_k^{(i)}(x) \rangle = \delta_{jk}$, $\forall j, k \in \mathbb{N}_0$. Without loss of generality, we restrict our attention to normalized polynomials.

Let $\mathbb{P}_n^{(i)}$ be the space of univariate polynomials of degree n or less. Let $\mathcal{P}_n^{(i)} : \mathcal{H}^{(i)} \rightarrow \mathbb{P}_n^{(i)}$ be an orthogonal projector onto this subspace. Then

$$\mathcal{P}_n^{(i)}(f) := \sum_{j=0}^n \langle f(x), \psi_j^{(i)}(x) \rangle \psi_j^{(i)}(x) = \sum_{j=0}^n f_j \psi_j^{(i)}(x), \quad (2.6)$$

with coefficients f_j defined according to the middle term above. The orthogonal projection operator yields the optimal L^2 approximation of f in $\mathbb{P}_n^{(i)}$, since any error is orthogonal to $\mathbb{P}_n^{(i)}$. The polynomials are dense in $\mathcal{H}^{(i)} = L^2(X^{(i)}, w^{(i)})$, so the polynomial expansion of any $f \in \mathcal{H}^{(i)}$ converges in the L^2 sense as $n \rightarrow \infty$ [4, 43]. If $f \in \mathcal{H}^{(i)}$, the coefficients must satisfy $\sum_{i=0}^{\infty} f_i^2 < \infty$.

Projections with finite degree n omit terms of the infinite series, thus incurring *truncation error*. We can write this error as

$$\|f - \mathcal{P}_n^{(i)}(f)\|_2^2 = \left\| \sum_{j=n+1}^{\infty} f_j \psi_j^{(i)} \right\|_2^2 = \sum_{j=n+1}^{\infty} f_j^2 < \infty. \quad (2.7)$$

Hence, we may reduce the truncation error to any desired level by increasing n , removing terms from the sum in (2.7) [4, 16].

2.5. Tensor polynomial chaos expansions. Let $\mathcal{H}^{(\mathbf{d})}$ be the tensor product of the Hilbert spaces defined above:

$$\mathcal{H}^{(\mathbf{d})} := \mathcal{H}^{(1)} \otimes \cdots \otimes \mathcal{H}^{(d)}. \quad (2.8)$$

$\mathcal{H}^{(\mathbf{d})}$ is a Hilbert space of real-valued functions on \mathbf{X} , with inner product given by the integral form in (2.4), $\langle f, g \rangle := \mathcal{I}^{(\mathbf{d})}(fg)$. A basis $\{\Psi_{\mathbf{i}} : \mathbf{i} \in \mathbb{N}_0^d\}$ for this space may be formed by taking the tensor product of univariate orthonormal polynomials in each dimension, such that

$$\Psi_{\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^d \psi_{i_j}^{(j)}(x^{(j)}) \quad (2.9)$$

and i_j is the j^{th} component of the multi-index \mathbf{i} .

Now consider the tensor product of one-dimensional polynomial projection operators, each of degree n_i :

$$\mathcal{P}_{\mathbf{n}}^{(\mathbf{d})} := \mathcal{P}_{n_1}^{(1)} \otimes \cdots \otimes \mathcal{P}_{n_d}^{(d)}. \quad (2.10)$$

This is an operator that projects elements of $\mathcal{H}^{(\mathbf{d})}$ onto a multivariate polynomial space whose truncation is given by the vector $\mathbf{n} = (n_1, \dots, n_d) \in \mathbb{N}_0^d$. This truncation is independent in each dimension, such that the range of the projector contains polynomials of degree up to n_i in each input $x^{(i)}$.

$$\mathcal{P}_{\mathbf{n}}^{(\mathbf{d})}(f) = \sum_{i_1=0}^{n_1} \cdots \sum_{i_d=0}^{n_d} \langle f \Psi_{\mathbf{i}} \rangle \Psi_{\mathbf{i}} \quad (2.11)$$

As in the one-dimensional case, the truncation induces error equal to the sum of the squares of the omitted coefficients, which we may similarly reduce to zero as $n_i \rightarrow \infty, \forall i$. The multivariate polynomial expansion also converges in an L^2 sense for any $f \in \mathcal{H}^{(\mathbf{d})}$.

3. One Dimensional and Full Tensor Approximations. The previous section introduced the two problems we explore throughout this paper: integration and projection onto polynomial spaces. Our next step is to develop useful (computable) approximation algorithms for these problems. To this end, we employ quadrature and pseudospectral approximation, respectively. Furthermore, we introduce criteria to describe when these approximations are precise.

3.1. General setting. Given a one-dimensional linear operator $\mathcal{L}^{(i)}$, we work with a convergent sequence of computable approximations, $\mathcal{L}_m^{(i)}$, such that

$$\|\mathcal{L}^{(i)} - \mathcal{L}_m^{(i)}\| \rightarrow 0 \text{ as } m \rightarrow \infty \quad (3.1)$$

in some appropriate norm. Taking the tensor product of these approximations provides an approximation to the full tensor operator:

$$\mathcal{L}^{(\mathbf{d})} \approx \mathcal{L}_{\mathbf{m}}^{(\mathbf{d})} := \mathcal{L}_{m_1}^{(1)} \otimes \cdots \otimes \mathcal{L}_{m_d}^{(d)}. \quad (3.2)$$

The level of the approximation may be individually selected in each dimension, so the tensor approximation is identified by a multi-index. As a consequence of the one-dimensional convergence, we have

$$\|\mathcal{L}^{(\mathbf{d})} - \mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}\| \rightarrow 0 \text{ as } \mathbf{m} \rightarrow \infty. \quad (3.3)$$

An important property of the approximation algorithms in which we are interested is whether they are *precise* for some inputs. We define the *accurate set* as the set of inputs for which the approximation is exact.

DEFINITION 3.1 (Accurate Sets). *For an operator \mathcal{L} and a corresponding approximation \mathcal{L}_m , define the accurate set as $\mathcal{A}(\mathcal{L}_m) := \{f : \mathcal{L}(f) = \mathcal{L}_m(f)\}$.*

It will also be useful (e.g., for quadrature) to consider when application of the operator to f^2 is precise.

DEFINITION 3.2 (Half Accurate Sets). *For an operator \mathcal{L} and a corresponding approximation \mathcal{L}_m , define the half accurate set $\mathcal{A}_2(\mathcal{L}_m) := \{f : \mathcal{L}(f^2) = \mathcal{L}_m(f^2)\}$.*

By linearity of the operator and its approximations, we can prove the following theorem relating the accurate sets of one-dimensional approximations and tensor approximations.

THEOREM 3.3. *If a tensor approximation $\mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}$ is constructed from one-dimensional approximations $\mathcal{L}_m^{(i)}$ with known accurate sets, then*

$$\mathcal{A}(\mathcal{L}_{m_1}^{(1)}) \otimes \cdots \otimes \mathcal{A}(\mathcal{L}_{m_d}^{(d)}) \subseteq \mathcal{A}(\mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}) \quad (3.4)$$

Proof. It is sufficient to show that the approximation is precise for an arbitrary input $f(\mathbf{x}) = f^{(1)}(x^{(1)})f^{(2)}(x^{(2)}) \cdots f^{(d)}(x^{(d)})$ where $f^{(i)}(x^{(i)}) \in \mathcal{A}(\mathcal{L}_{m_i}^{(i)})$, because we may extend it to sums of products by linearity:

$$\begin{aligned} \mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}(f^{(1)} \cdots f^{(d)}) &= \mathcal{L}_{m_1}^{(1)}(f^{(1)}) \otimes \cdots \otimes \mathcal{L}_{m_d}^{(d)}(f^{(d)}) \\ &= \mathcal{L}^{(1)}(f^{(1)}) \otimes \cdots \otimes \mathcal{L}^{(d)}(f^{(d)}) = \mathcal{L}^{(\mathbf{d})}(f). \end{aligned} \quad (3.5)$$

This replacement is by linearity of tensor products and the definition of accurate sets, proving the statement. \square

3.2. One-dimensional quadrature. Numerical quadrature approximates the action of an integral operator $\mathcal{I}^{(i)}$ with a weighted sum of point evaluations. For some family of quadrature rules, we write the “level m ” quadrature rule, comprised of $p^{(i)}(m) : \mathbb{N} \rightarrow \mathbb{N}$ points, as

$$\mathcal{I}^{(i)}(f) \approx \mathcal{Q}_m^{(i)}(f) := \sum_{j=1}^{p^{(i)}(m)} w_j^{(i)} f(x_j^{(i)}) \quad (3.6)$$

The choice of $p^{(i)}(m)$ depends on the quadrature family, as some rules only exist for certain numbers of points. For others, we tailor $p^{(i)}(m)$ depending on the desired properties, e.g., linear or exponential growth.

Many quadrature families are precise if f is a polynomial of a particular degree or less, which allows us to specify a part of the accurate set. This definition leads to the following lemma.

LEMMA 3.4. *For a one-dimensional quadrature rule with known polynomial accuracy $a^{(i)}(m)$, $\mathbb{P}_{a^{(i)}(m)} \subseteq \mathcal{A}(\mathcal{Q}_m^{(i)})$ and $\mathbb{P}_{\lfloor a^{(i)}(m)/2 \rfloor} \subseteq \mathcal{A}_2(\mathcal{Q}_m^{(i)})$.*

It is intuitive and useful in later sections to draw the accurate set, as in the following example:

EXAMPLE 3.5. *Figure 3.1 depicts the accurate set of a one-dimensional Gaussian quadrature rule with three points, $\mathcal{Q}_3^{(i)}$, which is accurate for polynomials up to fifth order. In this figure, the dots represent increasing polynomial orders of x_i . The solid box represents the extent of $\mathcal{A}(\mathcal{Q}_3^{(i)})$, hence only polynomials inside or on the box are correctly integrated in general. The dashed box depicts the half accurate set, $\mathcal{A}_2(\mathcal{Q}_3^{(i)})$, drawn as half the geometric size. This skips the step of rounding down, but is visually more intuitive and does not change the result.*

Most quadrature families are constructed so that $\mathcal{Q}_m^{(i)} \rightarrow \mathcal{I}^{(i)}$ as $m \rightarrow \infty$ for a wide variety of functions f . For this work, we rely on quadrature rules that exhibit polynomial accuracy of increasing order, which is sufficient to demonstrate convergence for functions in L^2 .

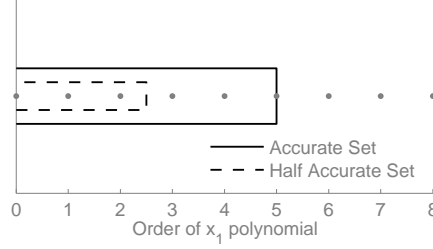


FIGURE 3.1. The accurate set $\mathcal{A}(\mathcal{Q}_3^1)$ and half accurate set $\mathcal{A}_2(\mathcal{Q}_3^1)$ of a three point Gaussian quadrature rule. Note that the vertical axis is meaningless here, as the figure is actually one dimensional.

3.3. Multi-indices. Before continuing, we must make a short diversion to multi-indices, which provide helpful notation when dealing with tensor problems. A multi-index is a vector $\mathbf{i} \in \mathbb{N}_0^d$. An important notion for multi-indices are *neighborhoods*.

DEFINITION 3.6 (Neighborhoods of multi-indices). A forward neighborhood of a multi-index \mathbf{k} is the multi-index set $n_f(\mathbf{k}) := \{\mathbf{k} + \mathbf{e}_i : \forall i \in \{1 \dots d\}\}$, where \mathbf{e}_i are the canonical unit vectors. The backward neighborhood of a multi-index \mathbf{k} is the multi-index set $n_b(\mathbf{k}) := \{\mathbf{k} - \mathbf{e}_i : \forall i \in \{1 \dots d\}, \mathbf{k} - \mathbf{e}_i \in \mathbb{N}_0^d\}$.

Smolyak algorithms rely on multi-index sets that are *admissible*.

DEFINITION 3.7 (Admissible multi-indices and multi-index sets). A multi-index \mathbf{k} is admissible with respect to a multi-index set \mathcal{K} if $n_b(\mathbf{k}) \subseteq \mathcal{K}$. A multi-index set \mathcal{K} is admissible if every $\mathbf{k} \in \mathcal{K}$ is admissible with respect to \mathcal{K} .

EXAMPLE 3.8. The two dimensional index set, $\mathcal{K} \subset \mathbb{N}_0^2$,

$$\mathcal{K} = \{(0, 0), (0, 1), (1, 0), (2, 0)\} \quad (3.7)$$

is admissible because the backward neighbors of all the multi-indices in the set \mathcal{K} are also in \mathcal{K} . For example, $n_b((0, 0)) = \{\}$ because its backward neighbors are not actually in \mathbb{N}_0^2 . Also, $n_b((1, 0)) = \{(0, 0)\}$, which is in the set. There are indices that are admissible with respect to \mathcal{K} , but not in \mathcal{K} : $(3, 0), (0, 2)$, or $(1, 1)$. Any of these may be added to \mathcal{K} to produce another admissible index set. The multi-indices $(3, 1), (4, 0)$, or $(4, 4)$ are inadmissible with respect to \mathcal{K} , so adding them to \mathcal{K} would produce an inadmissible multi-index set.

Two common admissible multi-index sets with simple geometric structure are *total order* multi-index sets and *full tensor* multi-index sets. We often encounter total order sets in the sparse grids literature and full tensor sets when dealing with tensor grids of points. The total order multi-index set \mathcal{K}_n^t comprises those multi-indices that lie within a d -dimensional simplex of side length n :

$$\mathcal{K}_n^t := \{\mathbf{k} \in \mathbb{N}_0^d : \|\mathbf{k}\|_1 \leq n\} \quad (3.8)$$

The full tensor multi-index set $\mathcal{K}_{\mathbf{n}}^f$ is the complete grid of indices bounded term-wise by a multi-index \mathbf{n} :

$$\mathcal{K}_{\mathbf{n}}^f := \{\mathbf{k} \in \mathbb{N}_0^d : \forall i \in \{1 \dots d\}, k_i < n_i\} \quad (3.9)$$

3.4. Tensor product quadrature. Using the linearity of integral operators, we can construct the *full tensor quadrature* approximation to a multi-dimensional integral $\mathcal{I}^{(d)}$ by taking the tensor product of quadrature operators:

$$\begin{aligned} \mathcal{Q}_{\mathbf{m}}^{(d)}(f) &:= \mathcal{Q}_{m_1}^{(1)} \otimes \dots \otimes \mathcal{Q}_{m_d}^{(d)}(f) \\ &= \sum_{\mathbf{j} \in \mathcal{J}} w_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}) \end{aligned} \quad (3.10)$$

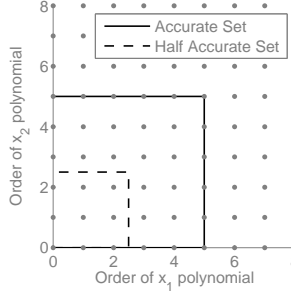


FIGURE 3.2. The quadrature rule accurate set $\mathcal{A}(\mathcal{Q}_{(3,3)}^{(2)})$ and half accurate set $\mathcal{A}_2(\mathcal{Q}_{(3,3)}^{(2)})$ for a two dimensional rule constructed from three point Gaussian quadrature rules.

where

$$\mathcal{J} := \mathcal{K}_{p^{(\mathbf{d})}(\mathbf{m})}^f \quad (3.11)$$

$$\mathbf{x}_{\mathbf{j}} := \left(x_{j_1}^{(1)}, \dots, x_{j_d}^{(d)} \right) \quad (3.12)$$

$$w_{\mathbf{j}} := \prod_{i=1}^d w_{j_i}^{(i)} \quad (3.13)$$

This quadrature rule is completely defined by the choice of one-dimensional quadrature families, and the nodes and weights are just combinations of the constituent one-dimensional rules. As a consequence of Theorem 3.3, we can describe the accurate set of the tensor algorithm.

COROLLARY 3.9. *The accurate set of a tensor quadrature rule constructed from constituent approximations with known accurate sets includes $\mathcal{A}(\mathcal{Q}_{m_1}^{(1)}) \otimes \dots \otimes \mathcal{A}(\mathcal{Q}_{m_d}^{(d)}) \subseteq \mathcal{A}(\mathcal{Q}_{\mathbf{m}}^{(\mathbf{d})})$.*

EXAMPLE 3.10. *Let $\mathcal{Q}_{(3,3)}^{(2)}$ be the two-dimensional full tensor quadrature rule constructed from three point Gaussian quadrature rules. Then $\mathbb{P}_5^{(1)} \otimes \mathbb{P}_5^{(2)} \subseteq \mathcal{A}(\mathcal{Q}_{(3,3)}^{(2)})$ and $\mathbb{P}_2^{(1)} \otimes \mathbb{P}_2^{(2)} \subseteq \mathcal{A}_2(\mathcal{Q}_{(3,3)}^{(2)})$. These accurate sets are depicted in Figure 3.2.*

While Corollary 3.9 specifies the natural extension of the accurate set to the tensor case, the accurate set may also include other functions that have more structure:

LEMMA 3.11. *Let f be separable, without loss of generality, with respect to its first input, i.e., $f(\mathbf{x}) = f'(x_1)f''(x_2, \dots, x_d)$. Further, let $\mathcal{I}^1(f') = 0$ and let $f'(x_1) \in \mathcal{A}(\mathcal{Q}_{m_1}^{(1)})$. Then $\mathcal{Q}_{\mathbf{m}}^{(\mathbf{d})}(f) = \mathcal{I}^{(\mathbf{d})}(f) = 0$ for any f'' . Hence, $f \in \mathcal{A}(\mathcal{Q}_{\mathbf{m}}^{(\mathbf{d})})$. This is a useful property arising from the linearity of integrals, which we use in our analysis of sparse problems.*

By the same argument as in the one-dimensional case, for functions $f \in L^2$ we have $\mathcal{Q}_{\mathbf{m}}^{(\mathbf{d})}(f) \rightarrow \mathcal{I}^{(\mathbf{d})}(f)$.

3.5. Pseudospectral approximation in one dimension. Pseudospectral approximation provides a practical non-intrusive algorithm by approximating a truncated polynomial chaos expansion with quadrature operators. Define the pseudospectral approximation in one dimension as

$$\mathcal{S}_m^{(i)}(f) := \sum_{j=0}^{q^{(i)}(m)} \mathcal{Q}_m^{(i)} \left(f \psi_j^{(i)} \right) \psi_j^{(i)}(x) \quad (3.14)$$

$$= \sum_{j=0}^{q^{(i)}(m)} \tilde{f}_j \psi_j^{(i)}(x) \quad (3.15)$$

where $q^{(i)}(m)$ is the polynomial truncation, to be specified shortly [4, 16]. Pseudospectral approximations are constructed around a level m quadrature rule, and may include as many terms in the sum as possible while maintaining accuracy. Assuming that $f \in L^2$, we can compute the L^2 error between the pseudospectral approximation and an exact projection onto the same polynomial space:

$$\left\| \mathcal{P}_{q^{(i)}(m)}^{(i)}(f) - \mathcal{S}_m^{(i)}(f) \right\|^2 = \left\| \sum_{j=0}^{q^{(i)}(m)} f_j \psi_j^{(i)} - \sum_{k=0}^{q^{(i)}(m)} \tilde{f}_k \psi_k^{(1)} \right\|_2^2 = \sum_{j=0}^{q^{(i)}(m)} (f_j - \tilde{f}_j)^2 \quad (3.16)$$

This quantity is the *aliasing error* [4, 16]. The error is non-zero because quadrature in general only approximates integrals, hence each \tilde{f}_i is an approximation of f_i . The pseudospectral operator also incurs truncation error, as before, which is orthogonal to the aliasing error. We can expand each approximate coefficient

$$\tilde{f}_j = \mathcal{Q}_m^{(i)} \left(f \psi_j^{(i)} \right) \quad (3.17)$$

$$= \sum_{k=0}^{\infty} f_k \mathcal{Q}_m^{(i)} \left(\psi_j^{(i)} \psi_k^{(i)} \right) \quad (3.18)$$

$$= \sum_{k=0}^{q^{(i)}(m)} f_k \mathcal{Q}_m^{(i)} \left(\psi_j^{(i)} \psi_k^{(i)} \right) + \sum_{l=q^{(i)}(m)+1}^{\infty} f_l \mathcal{Q}_m^{(i)} \left(\psi_j^{(i)} \psi_l^{(i)} \right) \quad (3.19)$$

The first step substitutes in the polynomial expansion of f , which we assume is convergent, and rearranges using linearity. The second step partitions the sum around the truncation of the pseudospectral expansion. Although $\langle \psi_j^{(i)}, \psi_k^{(i)} \rangle = \delta_{jk}$, we cannot assume in general that $\mathcal{Q}_m^{(i)} \left(\psi_j^{(i)} \psi_k^{(i)} \right) = \delta_{jk}$. Now substitute (3.19) back into the aliasing error expression:

$$\sum_{j=0}^{q^{(i)}(m)} (f_j - \tilde{f}_j)^2 = \sum_{j=0}^{q^{(i)}(m)} \left(f_j - \sum_{k=0}^{q^{(i)}(m)} f_k \mathcal{Q}_m^{(i)} \left(\psi_j^{(i)} \psi_k^{(i)} \right) - \sum_{l=q^{(i)}(m)+1}^{\infty} f_l \mathcal{Q}_m^{(i)} \left(\psi_j^{(i)} \psi_l^{(i)} \right) \right)^2 \quad (3.20)$$

This form reveals the intimate link between the accuracy of pseudospectral approximations and the polynomial accuracy of quadrature rules. All aliasing is attributed to the inability of the quadrature rule to determine the orthonormality of the basis polynomials, causing one coefficient to corrupt another. The contribution of the first two parenthetical terms on the right of (3.20) is called *internal aliasing* while the third term is called *external aliasing*. Internal aliasing is due to inaccuracies in $\mathcal{Q}(\psi_j^{(i)} \psi_k^{(i)})$ when both $\psi_j^{(i)}$ and $\psi_k^{(i)}$ are included in the expansion, while external aliasing occurs when only one of these polynomials is included in the expansion. For many practical quadrature rules—and for all those used in this work—whenever $(\psi_j^{(i)} \psi_k^{(i)}) \notin \mathcal{A}(\mathcal{Q})$ and hence the discrete inner product is not zero, $\left\| \mathcal{Q}(\psi_j^{(i)} \psi_k^{(i)}) \right\|$ is $\mathcal{O}(1)$ [34]. This provides an important order-of-magnitude estimate of the error.

Both types of aliasing errors are driven to zero by sufficiently powerful quadrature, but we are left with a practical trade-off. The obvious option is to select the smallest quadrature level that sets the internal aliasing to zero, i.e., that allows the pseudospectral operator to precisely recover functions that actually lie in the space spanned by polynomials included in the truncated expansion. We refer to this space as the *range* of the pseudospectral projection operator.

Furthermore, recall that because $\sum f_i^2 < \infty$, the high-order coefficients must in general decay. Aliasing error is proportional to coefficient magnitude because $\left\| \mathcal{Q}(\psi_k^{(i)} \psi_l^{(i)}) \right\|$ is $\mathcal{O}(1)$. Therefore, we expect that for a pseudospectral method to accurately approximate a particular f , the possible errors resulting from internal aliasing are much larger than those from external

aliasing. Indeed, we see in practice that any internal aliasing can lead to hopeless corruption of the approximation.

Analysis of (3.20) yields a theorem defining when the pseudospectral projection operator has no internal aliasing and thus is *precise* on its range space:

THEOREM 3.12. *The internal aliasing of $\mathcal{S}_m^{(i)}$ is zero if $\forall j, k \leq q^{(i)}(m), (\psi_j^{(i)} \psi_k^{(i)}) \in \mathcal{A}(\mathcal{Q}_m^{(i)})$, or equivalently, $\forall j \leq q^{(i)}(m), \psi_j^{(i)} \in \mathcal{A}_2(\mathcal{Q}_m^{(i)})$. Thus $\mathbb{P}_{q^{(i)}(m)}^{(i)} \subseteq \mathcal{A}(\mathcal{S}_m^{(i)})$.*

The most efficient choice is to include as many basis terms as possible without introducing internal aliasing error.

LEMMA 3.13. *The conditions of Theorem 3.12 are satisfied if $q^{(i)}(m) = \text{floor}(a^{(i)}(m)/2)$ when $\mathcal{Q}_m^{(i)}$ has polynomial accuracy $a^{(i)}(m)$.*

EXAMPLE 3.14. *Revisit Figure 3.1, which shows accurate sets for a three-point Gaussian quadrature rule. The dashed box, $\mathcal{A}_2(\mathcal{Q}_3^1)$, shows all the terms that can be included in the pseudospectral approximation constructed from this quadrature rule without inducing internal aliasing—that is, terms which are precisely recovered by the operator. In general, an m -point Gaussian quadrature rule allows one to compute polynomial approximations up to order $m - 1$.*

Given that Theorem 3.12 is satisfied, we wish to show that the pseudospectral approximation converges to the true function, where the magnitude of the error is as follows:

$$\|f - \mathcal{S}_m^{(i)}(f)\|_2^2 = \sum_{j=0}^{q^{(i)}(m)} \left(\sum_{k=q^{(i)}(m)+1}^{\infty} f_k \mathcal{Q}(\psi_j^{(i)} \psi_k^{(i)}) \right)^2 + \sum_{l=q^{(i)}(m)+1}^{\infty} f_l^2. \quad (3.21)$$

The two terms on right hand side comprise the external aliasing and the truncation error, respectively. We already know that the truncation error goes to zero as $q^{(i)}(m) \rightarrow \infty$. The external aliasing also vanishes for functions $f \in L^2$ as the truncated portion of f likewise decreases [34]. In the case of Gaussian quadrature rules, a link to interpolation provides precise rates for the convergence of the pseudospectral operator based on the regularity of f [4].

3.6. Full tensor pseudospectral approximation in d dimensions. As with quadrature algorithms, pseudospectral approximation in one dimension is directly extensible to full tensor problems by taking the tensor product of the one-dimensional approximations,

$$\mathcal{S}_{\mathbf{m}}^{(\mathbf{d})} := \mathcal{S}_{m_1}^{(1)} \otimes \cdots \otimes \mathcal{S}_{m_d}^{(d)} = \sum_{\mathbf{k} \in \mathcal{K}_{\mathbf{q}(\mathbf{m})}^f} \mathcal{Q}_{\mathbf{m}}^{(\mathbf{d})}(f \Psi_{\mathbf{k}}) \Psi_{\mathbf{k}}(\mathbf{x}) \quad (3.22)$$

where $\mathbf{q}(\mathbf{m}) = (q^{(1)}(m_1), \dots, q^{(d)}(m_d))$. This expansion contains all basis functions up to and including $\Psi_{\mathbf{q}(\mathbf{m})}$. As in the one-dimensional case, error in the approximation can be separated into truncation, internal aliasing, and external aliasing. We may again apply Theorem 3.3 to derive the accuracy of the tensor product algorithm.

COROLLARY 3.15. *The accurate set of a tensor product pseudospectral approximation operator constructed from constituent approximations with known accurate sets includes $\mathcal{A}(\mathcal{S}_{m_1}^{(1)}) \otimes \cdots \otimes \mathcal{A}(\mathcal{S}_{m_d}^{(d)}) \subseteq \mathcal{A}(\mathcal{S}_{\mathbf{m}}^{(\mathbf{d})})$.*

We conclude that the full tensor approximation has no internal aliasing.

EXAMPLE 3.16. *Figure 3.2 shows the accurate and half-accurate sets of $\mathcal{Q}_{(3,3)}^2$, indicating which terms may be included in the pseudospectral expansion without introducing internal aliasing error.*

The same analysis as in the one-dimensional case allows us to conclude that the d -dimensional pseudospectral approximation converges to the true function.

4. Smolyak Algorithms. Thus far, we have developed polynomial approximations of multivariate functions by taking tensor products of one-dimensional pseudospectral operators. Smolyak algorithms avoid the exponential cost of full tensor products by assuming that the input

dimensions are not fully coupled, and then using a telescoping sum to blend different full tensor approximations.

EXAMPLE 4.1. Assume that $f(x, y) = x^7 + y^7 + x^3y$. To construct a polynomial expansion with both the x^7 and y^7 terms, a full tensor pseudospectral algorithm would estimate all the polynomial terms up to x^7y^7 , because tensor algorithms fully couple the dimensions. This mixed term is costly, requiring, for example, an 8×8 point grid for Gaussian quadratures. The individual terms can be had much more cheaply, using 8×1 , 1×8 , and 4×2 grids, respectively. Smolyak algorithms help realize such savings in practice.

4.1. General Smolyak algorithms. As in Section 3, assume that we have for every dimension $i = 1 \dots d$ a convergent sequence $\mathcal{L}_k^{(i)}$ of approximations to a one-dimensional linear operator $\mathcal{L}^{(i)}$, such that $\|\mathcal{L}^{(i)} - \mathcal{L}_k^{(i)}\| \rightarrow 0$ as $k \rightarrow \infty$. Let \mathcal{L} denote the collection of these sequences over all the dimensions. For any i , we may write the exact or “true” operator as the telescoping series

$$\mathcal{L}^{(i)} = \sum_{k=0}^{\infty} \mathcal{L}_k^{(i)} - \mathcal{L}_{k-1}^{(i)}. \quad (4.1)$$

Define the difference operators

$$\Delta_0^{(i)} := \mathcal{L}_0^{(i)} = 0, \quad (4.2)$$

$$\Delta_n^{(i)} := \mathcal{L}_n^{(i)} - \mathcal{L}_{n-1}^{(i)}. \quad (4.3)$$

These allow us to rewrite the telescoping sum as

$$\mathcal{L}^{(i)} = \sum_{k=0}^{\infty} \Delta_k^{(i)}. \quad (4.4)$$

Now we may write the tensor product of the exact operators as the tensor product of the telescoping sums, and interchange the product and sum:

$$\mathcal{L}^{(1)} \otimes \dots \otimes \mathcal{L}^{(d)} = \sum_{k_1=0}^{\infty} \Delta_{k_1}^{(1)} \otimes \dots \otimes \sum_{k_d=0}^{\infty} \Delta_{k_d}^{(d)} \quad (4.5)$$

$$= \sum_{\mathbf{k}=0}^{\infty} \Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_d}^{(d)} \quad (4.6)$$

Smolyak’s idea is to approximate the tensor product operator with truncations of this sum [31]:

$$A(\mathcal{K}, d, \mathcal{L}) := \sum_{\mathbf{k} \in \mathcal{K}} \Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_d}^{(d)}. \quad (4.7)$$

We refer to the multi-index set \mathcal{K} as the *Smolyak multi-index set*, and it must be admissible for the sum to telescope correctly. Smolyak specifically suggested truncating with a total order multi-index set, which is the most widely studied choice. However, we can compute the approximation with any admissible multi-index set. Although the expression above is especially clean, it is not the most useful form for computation. We can reorganize the terms of (4.7) to construct a weighted sum of the tensor operators:

$$A(\mathcal{K}, d, \mathcal{L}) = \sum_{\mathbf{k} \in \mathcal{K}} c_{\mathbf{k}} \mathcal{L}_{k_1}^{(1)} \otimes \dots \otimes \mathcal{L}_{k_d}^{(d)}, \quad (4.8)$$

where $c_{\mathbf{k}}$ are integer *Smolyak coefficients* computed from the combinatorics of the difference formulation. One can compute the coefficients through a simple iteration over the index set and

use (4.7) to determine which full tensor rules are incremented or decremented. In general, these coefficients are non-zero near the leading surface of the Smolyak multi-index set, reflecting the mixing of the most accurate constituent full tensor approximations.

If each sequence of one-dimensional operators converges, then the Smolyak approximation converges to the tensor product of exact operators as $\mathcal{K} \rightarrow \mathbb{N}_0^d$. For the isotropic simplex index set, some precise rates of convergence are known with respect to the side length of the simplex [36, 37, 35, 28, 30]. Although general admissible Smolyak multi-index sets are difficult to study theoretically, they allow detailed customization to the anisotropy of a particular function.

4.2. Accuracy of Smolyak algorithms. In the one-dimensional and full tensor settings, we have characterized approximation algorithms through their accurate sets—those inputs for which the algorithm is precise. This section shows that if the constituent one-dimensional approximations have nested accurate sets, Smolyak algorithms are the ideal blending of different full tensor approximations from the perspective of accurate sets; that is, the accurate set of the Smolyak algorithm contains the union of the accurate sets of the component full tensor approximations. This result will facilitate subsequent analysis of sparse quadrature and pseudospectral approximation algorithms.

THEOREM 4.2. *Let $A(\mathcal{K}, d, \mathcal{L})$ be a Smolyak algorithm composed of linear operators with nested accurate sets. That is, let \mathcal{K} be admissible and $m \leq m'$ imply $\mathcal{A}(\mathcal{L}_m^{(i)}) \subseteq \mathcal{A}(\mathcal{L}_{m'}^{(i)})$ for $i = 1 \dots d$. Then the accurate set of $A(\mathcal{K}, d, \mathcal{L})$ contains*

$$\mathcal{A}(A(\mathcal{K}, d, \mathcal{L})) \supseteq \bigcup_{\mathbf{k} \in \mathcal{K}} \mathcal{A}(\mathcal{L}_{k_1}^{(1)} \otimes \dots \otimes \mathcal{L}_{k_d}^{(d)}) \quad (4.9)$$

$$\supseteq \bigcup_{\mathbf{k} \in \mathcal{K}} \mathcal{A}(\mathcal{L}_{k_1}^{(1)}) \otimes \dots \otimes \mathcal{A}(\mathcal{L}_{k_d}^{(d)}). \quad (4.10)$$

Our proof closely follows the framework provided by Novak and Ritter, but includes a generalization to arbitrary Smolyak multi-index sets [24, 25, 2]. We begin by introducing notation to incrementally build a multi-index set dimension by dimension.

DEFINITION 4.3. *For a multi-index set \mathcal{K} of dimension d , let the restriction of the multi-indices to the first i dimensions be $\mathcal{K}^{(i)} := \{\mathbf{k}_{1:i} = (k_1, \dots, k_i) : \mathbf{k} \in \mathcal{K}\}$. Furthermore, define subsets of \mathcal{K} based on the i^{th} element of the multi-indices, $\mathcal{K}_j^{(i)} := \{\mathbf{k}_{1:i} : \mathbf{k} \in \mathcal{K}^{(i)} \text{ and } k_{i+1} = j\}$. These sets are nested, $\mathcal{K}_j^{(i)} \supseteq \mathcal{K}_{j+1}^{(i)}$, because \mathcal{K} is admissible. Also let k_i^{\max} denote the maximum value of the i^{th} component of the multi-indices in the set \mathcal{K} .*

Using this notation, one can construct \mathcal{K} inductively,

$$\mathcal{K}^{(1)} = \{1, \dots, k_1^{\max}\} \quad (4.11)$$

$$\mathcal{K}^{(i)} = \bigcup_{j=1}^{k_i^{\max}} \mathcal{K}_j^{(i-1)} \otimes j, \quad i = 2 \dots d. \quad (4.12)$$

Proof of Theorem 4.2. It is sufficient to prove that the Smolyak operator is precise for an arbitrary f with tensor structure, $f = f_1 \otimes \dots \otimes f_d$. Suppose there exists a \mathbf{k}^* such that $f \in \mathcal{A}(\mathcal{L}_{\mathbf{k}^*}^{(d)})$. We will show that if \mathcal{K} is an admissible multi-index set containing \mathbf{k}^* , then $A(\mathcal{K}, d, \mathcal{L})$ is precise on f . We do so by induction on the dimension i of the Smolyak operator and the function.

First, consider the $i = 1$ case. $A(\mathcal{K}^{(1)}, 1, \mathcal{L}) = \mathcal{L}_{k_1^{\max}}^{(1)}$, where $k_1^{\max} \geq k_1^*$. Hence $\mathcal{A}(A(\mathcal{K}^{(1)}, 1, \mathcal{L})) = \mathcal{A}(\mathcal{L}_{k_1^{\max}}^{(1)})$.

For the induction step, we construct the $(i + 1)$ -dimensional Smolyak operator in terms of the i -dimensional operator:

$$A(\mathcal{K}^{(i+1)}, i + 1, \mathcal{L}) = \sum_{j=1}^{k_{i+1}^{\max}} A(\mathcal{K}_j^{(i)}, i, \mathcal{L}) \otimes (\mathcal{L}_j^{(i+1)} - \mathcal{L}_{j-1}^{(i+1)}). \quad (4.13)$$

This sum is over increasing levels of accuracy in the $i + 1$ dimension. We know the level required for the approximate operator to be precise in this dimension; this may be expressed as

$$\mathcal{L}_j^{(i+1)}(f_{i+1}) = \mathcal{L}_{j-1}^{(i+1)}(f_{i+1}) = \mathcal{L}^{(i+1)}(f_{i+1}) \text{ when } j - 1 \geq k_{i+1}^*. \quad (4.14)$$

Therefore the sum (4.13) can be truncated at the k_{i+1}^* term, as the differences of higher terms are zero when applied to f :

$$A(\mathcal{K}^{(i+1)}, i + 1, \mathcal{L}) = \sum_{j=1}^{k_{i+1}^*} A(\mathcal{K}_j^{(i)}, i, \mathcal{L}) \otimes (\mathcal{L}_j^{(i+1)} - \mathcal{L}_{j-1}^{(i+1)}). \quad (4.15)$$

Naturally, $\mathbf{k}_{1:i}^* \in \mathcal{K}_{k_{i+1}^*}^{(i)}$. By nestedness, $\mathbf{k}_{1:i}^*$ is also contained in $\mathcal{K}_j^{(i)}$ for $j \leq k_{i+1}^*$. The induction hypothesis then guarantees

$$f_1 \otimes \cdots \otimes f_i \in \mathcal{A}(A(\mathcal{K}_j^{(i)}, i, \mathcal{L})), \quad \forall j \leq k_{i+1}^*. \quad (4.16)$$

Applying the $(i + 1)$ -dimensional Smolyak operator to the truncated version of f yields

$$\begin{aligned} & A(\mathcal{K}^{(i+1)}, i + 1, \mathcal{L})(f_1 \otimes \cdots \otimes f_{i+1}) \\ &= \sum_{j=1}^{k_{i+1}^*} A(\mathcal{K}_j^{(i)}, i, \mathcal{L})(f_1 \otimes \cdots \otimes f_i) \otimes (\mathcal{L}_j^{(i+1)} - \mathcal{L}_{j-1}^{(i+1)})(f_{i+1}). \end{aligned} \quad (4.17)$$

Since each of the i -dimensional Smolyak algorithms is precise, by the induction hypothesis, we replace them with the true operators and rearrange by linearity to obtain

$$\begin{aligned} A(\mathcal{K}^{(i+1)}, i + 1, \mathcal{L})(f_1 \otimes \cdots \otimes f_{i+1}) &= \mathcal{L}^{(i)}(f_1 \otimes \cdots \otimes f_i) \otimes \sum_{j=1}^{k_{i+1}^*} (\mathcal{L}_j^{(i+1)} - \mathcal{L}_{j-1}^{(i+1)})(f_{i+1}) \\ &= \mathcal{L}^{(i)}(f_1 \otimes \cdots \otimes f_i) \otimes \mathcal{L}_{k_{i+1}^*}^{(i+1)}(f_{i+1}). \end{aligned} \quad (4.18) \quad (4.19)$$

The approximation in the $i + 1$ dimension is exactly of the level needed to be precise on the $(i + 1)^{\text{th}}$ component of f . Then (4.19) becomes

$$\mathcal{L}^{(i)}(f_1 \otimes \cdots \otimes f_i) \otimes \mathcal{L}^{(i+1)}(f_{i+1}) = \mathcal{L}^{(i+1)}(f_1 \otimes \cdots \otimes f_{i+1}) \quad (4.20)$$

Thus the Smolyak operator is precise for f , and the claim is proven. \square

4.3. Smolyak quadrature. We recall the most familiar use of Smolyak algorithms, sparse quadrature. Following the formalism of Section 4.1, consider a family of one-dimensional quadrature rules $(\mathcal{Q}_k^{(i)})$ in each dimension $i = 1 \dots d$; denote these rules by \mathcal{Q} . We define $\mathcal{Q}_0^{(i)} := 0$ as required for the Smolyak construction. A Smolyak quadrature algorithm in d dimensions is a weighted sum of full tensor quadrature rules:

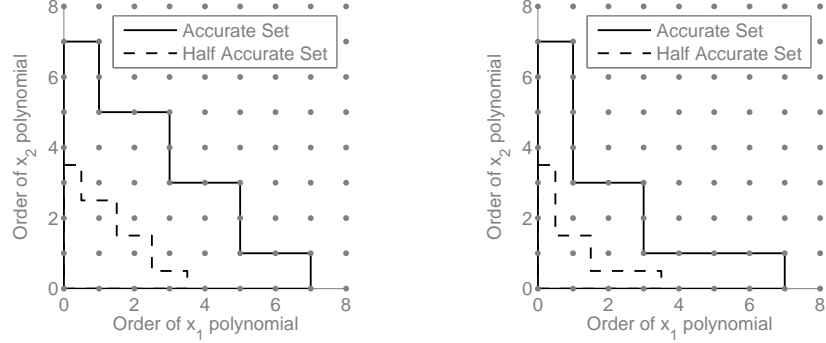
$$A(\mathcal{K}, d, \mathcal{Q}) = \sum_{\mathbf{k} \in \mathcal{K}} c_{\mathbf{k}} \mathcal{Q}_{\mathbf{k}}^{(d)} \quad (4.21)$$

The set of functions that are integrated correctly by a Smolyak quadrature algorithm is described as a corollary of Theorem 4.2.

COROLLARY 4.4. *For a sparse quadrature rule satisfying the hypotheses of Theorem 4.2 (that is, composed of one-dimensional quadrature rules that have nested accurate sets), we have*

$$\mathcal{A}(A(\mathcal{K}, d, \mathcal{Q})) \supseteq \bigcup_{\mathbf{k} \in \mathcal{K}} \mathcal{A}(\mathcal{Q}_{\mathbf{k}}^{(d)}) \quad (4.22)$$

FIGURE 4.1. The accurate set diagram for two Smolyak quadrature rules, and the corresponding basis for a Smolyak pseudospectral estimate. $\mathcal{A}(\mathcal{Q})$ is shown in a solid line, $\mathcal{A}_2(\mathcal{Q})$ is the dashed line.



(a) The accurate set for a four level Smolyak quadrature in two dimensions, based on *linear* growth Gaussian quadrature rules.

(b) The accurate set for a three level Smolyak quadrature in two dimensions, based on *exponential* growth Gaussian quadrature rules.

Additionally, other separable functions may be precisely integrated.

THEOREM 4.5. *If f is separable such that $f = f_1(x_1)f'(x_2, \dots, x_d)$ and $\mathcal{I}^{(1)}(f_1) = 0$, and if for every full tensor rule $\mathbf{k} \in \mathcal{K}$ we have $f_1 \in \mathcal{A}(\mathcal{Q}_{k_1}^{(1)})$, then $f \in \mathcal{A}(\mathcal{A}(\mathcal{K}, d, \mathcal{Q}))$.*

Proof. If every full tensor rule in the Smolyak algorithm correctly computes that f_1 integrates to zero, then their weighted sum is necessarily zero. \square

We conclude by illustrating accurate sets for some sparse quadrature rules.

EXAMPLE 4.6. *Figure 4.1 shows the accurate set for two instances of Smolyak quadrature, constructed from one dimensional Gaussian rules with different growth rates $p^{(1)}(m)$. The accurate sets look like steps, as they are the superposition of rectangular full tensor accurate sets.*

4.4. Smolyak pseudospectral approximation. Applying Smolyak's algorithm to pseudospectral approximation operators yields a sparse algorithm that converges under similar conditions as the one-dimensional operators it is constructed from. This algorithm is written

$$A(\mathcal{K}, d, \mathcal{S}) = \sum_{\mathbf{k} \in \mathcal{K}} c_{\mathbf{k}} \mathcal{S}_{\mathbf{k}}^{(d)}. \quad (4.23)$$

REMARK 4.7. *The Smolyak algorithm is a sum of different full tensor pseudospectral approximations, where each approximation is built around the polynomial accuracy of a single full tensor quadrature rule. It is not naturally expressed as a set of formulas for the polynomial coefficients, because different approximations include different polynomials.*

LEMMA 4.8. *When the one-dimensional rules are constructed to satisfy Lemma 3.13, the term $\Psi_{\mathbf{j}}$ is included in the Smolyak approximation if and only if $\exists \mathbf{k} \in \mathcal{K} : \Psi_{\mathbf{j}} \in \mathcal{A}_2(\mathcal{Q}_{\mathbf{k}}^{(d)})$. Here, $\mathcal{Q}_{\mathbf{k}}^{(d)}$ is a full tensor quadrature rule corresponding to the quadrature rule used by the full tensor pseudospectral approximation $\mathcal{S}_{\mathbf{k}}^{(d)}$.*

EXAMPLE 4.9. *Returning to Figure 4.1, the half accurate set again depicts the polynomial terms included in the corresponding Smolyak pseudospectral approximation.*

In principle, we could stop our analysis here, as the convergence of the sparse algorithm is assured by the properties of Smolyak's algorithm. However, for the purposes of comparing to direct quadrature, we can provide two theorems that characterize the internal aliasing and

external aliasing in more detail. Specific examples of internal and external behavior are given in the next section. The internal aliasing result follows as a corollary of Theorem 4.2.

COROLLARY 4.10. *If the constituent one-dimensional pseudospectral rules are consistent with Theorem 3.12, then the resulting Smolyak pseudospectral algorithm has no internal aliasing error. In other words, the Smolyak pseudospectral approximation is precise on its range.*

THEOREM 4.11. *Let $\Psi_{\mathbf{j}}$ be a polynomial term included in the expansion provided by the Smolyak algorithm $A(\mathcal{K}, d, \mathcal{S})$, and let $\Psi_{\mathbf{j}'}$ be a polynomial term not included in the expansion. There is no external aliasing of $\Psi_{\mathbf{j}'}$ onto $\Psi_{\mathbf{j}}$ if either (a) for any dimension i , $j'_i < j_i$ or (b) $\exists \mathbf{k} \in \mathcal{K}$ such that $\Psi_{\mathbf{j}}$ is included in the expansion provided by $\mathcal{S}_{\mathbf{k}}^{(d)}$ and $\Psi_{\mathbf{j}'}\Psi_{\mathbf{j}} \in \mathcal{A}(\mathcal{Q}_{\mathbf{k}}^{(d)})$, where \mathcal{Q} corresponds to the quadrature rules of \mathcal{S} .*

Proof. If condition (a) is satisfied, then $\Psi_{\mathbf{j}}$ and $\Psi_{\mathbf{j}'}$ are orthogonal in dimension i , and hence that inner product is zero. Every quadrature rule that computes the coefficient of $\Psi_{\mathbf{j}}$ is accurate for polynomials of at least order $2j_i$. Since $j'_i + j_i < 2j_i$, every rule that computes the coefficient can determine the orthogonality, and therefore there is no aliasing. If condition (b) is satisfied, then the result follows from the cancellations the Smolyak algorithm exploits, as seen in the proof of Theorem 4.2. \square

The properties stated in these theorems are extremely useful. First, any Smolyak pseudospectral algorithm, regardless of the admissible Smolyak multi-index set used, has no internal aliasing. Second, there is external aliasing, as we expect, but the algorithm uses basis orthogonality to limit which external coefficients can alias onto an included coefficient. Hence, the Smolyak construction is a useful approximation, in that we can tailor it to perform the desired amount of work and achieve reliable approximations of the selected coefficients. Computing an accurate approximation of the function only requires including enough terms so that the truncation and external aliasing errors are small.

5. Comparing Direct Quadrature to Smolyak Pseudospectral Approximation.

The current UQ literature often suggests a *direct quadrature* approach for constructing polynomial chaos expansions [40, 20, 7, 17]. In this section, we describe this approach and show that, in comparison to a true Smolyak algorithm, it is inaccurate or inefficient in almost all cases.

5.1. Direct quadrature polynomial expansions. At first glance, direct quadrature is quite simple. First, choose a multi-index set \mathcal{J} to define a truncated polynomial expansion:

$$f \approx \sum_{\mathbf{j} \in \mathcal{J}} \tilde{f}_{\mathbf{j}} \Psi_{\mathbf{j}}. \quad (5.1)$$

The index set \mathcal{J} is typically admissible, but need not be. Second, select any d -dimensional quadrature rule $\mathcal{Q}^{(d)}$, and estimate every coefficient as:

$$\tilde{f}_{\mathbf{j}} = \mathcal{Q}^{(d)}(f \Psi_{\mathbf{j}}). \quad (5.2)$$

Unlike the Smolyak approach, we are left to choose \mathcal{J} and $\mathcal{Q}^{(d)}$ independently, giving the appearance of flexibility. We are interested in selecting \mathcal{Q} and \mathcal{J} identically to the Smolyak approach, as in Lemma 4.8.

5.2. Internal aliasing in direct quadrature. As before, we define internal aliasing as the error incurred when one coefficient in the expansion aliases onto another because the included basis functions are not numerically orthogonal.

THEOREM 5.1. *For a multi-index set \mathcal{J} and a quadrature rule $\mathcal{Q}^{(d)}$, the corresponding direct quadrature polynomial expansion has no internal aliasing if and only if for every $\mathbf{j}, \mathbf{j}' \in \mathcal{J}$, $\Psi_{\mathbf{j}}\Psi_{\mathbf{j}'}$ is correctly integrable, that is, lies in $\mathcal{A}(\mathcal{Q}^{(d)})$.*

This theorem follows directly from the definition of accurate sets. We can immediately conclude that for any basis set \mathcal{J} , there is some quadrature rule sufficiently powerful to avoid internal aliasing error. In practice, however, this rule may not be a desirable one.

EXAMPLE 5.2. *Assume that for some function with two inputs, we wish to include the polynomial basis terms $(a, 0)$ and $(0, b)$. By Theorem 5.1, the product of these two terms must*

be in the accurate set; hence, the quadrature must include at least a full tensor rule of accuracy (a, b) . Although we have not asked for any coupling, direct quadrature must assume full coupling of the problem in order to avoid internal aliasing.

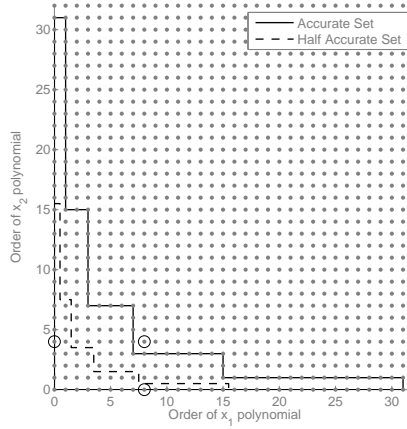
Thus we reach the surprising conclusion that direct quadrature is essentially incompatible with actually using sparse grid evaluations. For most sparse quadrature rules, we cannot include as many terms as the Smolyak pseudospectral approach without incurring internal aliasing, because the quadrature is not powerful enough in the mixed dimensions.

EXAMPLE 5.3. Let \mathbf{X} be the two-dimensional domain $[-1, 1]^2$. Select a uniform weight function, which corresponds to Legendre polynomials and Gauss-Legendre quadrature. Let $f(x, y) = \psi_0(x)\psi_4(y)$. Use an exponential growth rule for the quadrature, such that $p^{(i)}(m) = 2^{m-1}$. Select a sparse quadrature rule based on a total order multi-index set $\mathcal{Q}_{K_5^1}^2$. Figure 5.1 shows the accurate set of this Smolyak quadrature rule (solid line) along with its half-accurate set (dashed line), which encompasses all the terms in the direct quadrature PCE.

Consider the $\mathbf{j} = (8, 0)$ polynomial, which is in the half-accurate set. The product of the $(0, 4)$ and $(8, 0)$ polynomial terms is $(8, 4)$, which is not within the accurate set of the sparse rule. Hence, $(0, 4)$ aliases onto $(8, 0)$ because this quadrature rule has limited accuracy in the mixed dimensions.

Using both the Smolyak pseudospectral and direct quadrature methods, we numerically computed the polynomial expansion for this example. The resulting coefficients are shown in Figure 5.2. Even though the two methods use the same information and project f onto the same basis, the Smolyak result has no internal aliasing while direct quadrature shows significant internal aliasing. Although both methods correctly compute the $(0, 4)$ coefficient, direct quadrature shows aliasing on $(8, 0)$ as predicted, and also on $(10, 0)$, $(12, 0)$, and $(14, 0)$. This polynomial approximation is so drastically corrupted as to be completely useless. Alternating terms are computed correctly because of the parity of the functions.

FIGURE 5.1. The accurate set and polynomials included in the direct quadrature construction from Example 5.3.

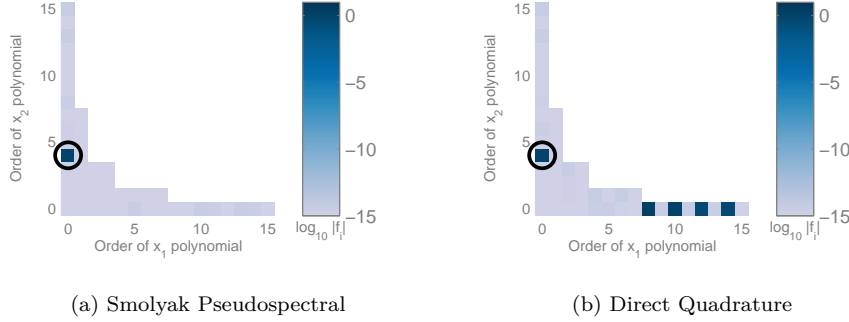


This example depicts a dangerous type of internal aliasing: a low order term, which we expect to be large, corrupts many other terms, which we would ordinarily expect to be much smaller.

We can understand this phenomena geometrically, in that whenever the accurate set \mathcal{A} is concave by more than unit steps, then the product of some pair of polynomials in \mathcal{A}_2 (drawn as the sum of the points), does not lie in \mathcal{A} . Figure 5.1 shows that \mathcal{A} is quite concave, leading to the observed errors. Forbidding all sparse quadrature rules with concave accurate sets—which imply very weak coupling between inputs—is quite restrictive! This generally excludes every nested quadrature rule, or any super-linear growth in the number of points per level.

REMARK 5.4. Sparse quadrature rules constructed from Gaussian quadrature rules with

FIGURE 5.2. Numerical results for Example 5.3; each color square indicates the log of the coefficient magnitude for the basis function at that position. The circle identifies the correct non-zero coefficient.



$p^{(i)}(m) = m$, truncated according to an isotropic total-order multi-index set, do not have internal aliasing. Although the accurate set has a stair-step shape of size two, this shape is “convex” enough.

The results and analysis in this section underpin one of key conclusions of this work: direct quadrature strategies fail to accurately compute polynomial expansions for most sparse quadrature rules, if one truncates the polynomial basis similarly to the Smolyak approach. It is possible to select a sufficiently small polynomial basis to avoid internal aliasing, but this approach requires drastic conservatism that could easily be avoided with a Smolyak pseudospectral approximation.

5.3. External aliasing. The difference in external aliasing between direct quadrature and Smolyak pseudospectral approximation is much less severe. Both methods exhibit external aliasing from terms far outside the estimated set. And since they are constructed from similar constituent one-dimensional quadrature rules, the aliasing is of similar magnitude when it occurs. The fundamental theorem is quite similar to Theorem 4.11.

THEOREM 5.5. *For a multi-index set \mathcal{J} and a quadrature rule $\mathcal{Q}^{(d)}$, the corresponding direct quadrature PCE approximation has no external aliasing between an included term $\Psi_{\mathbf{j}}$ and a non-included term $\Psi_{\mathbf{j}'}$ if $\Psi_{\mathbf{j}'}, \Psi_{\mathbf{j}} \in \mathcal{A}(\mathcal{Q}^{(d)})$.*

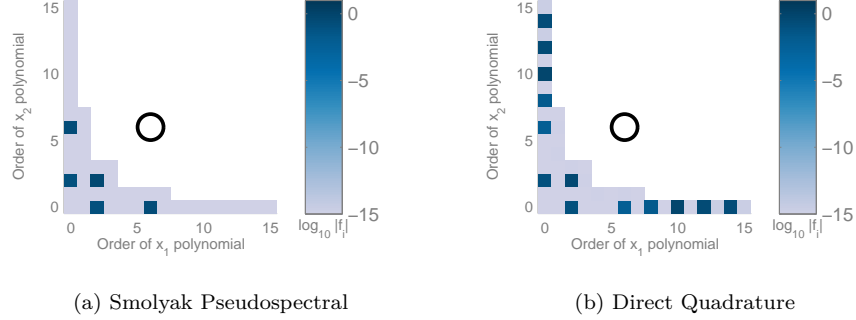
Yet the two methods perform differently because of their behavior on separable functions. The Smolyak pseudospectral algorithm gains a condition that prevents external aliasing as a result of Theorem 4.5 and thus it has strictly less external aliasing in general.

EXAMPLE 5.6. *If we repeat Example 5.3 but choose f to be a polynomial outside the approximation basis, $f = \psi_6(x)\psi_6(y)$, we obtain the results in Figure 5.3. Now every non-zero coefficient is the result of external aliasing. The Smolyak pseudospectral approach incorrectly constructs several coefficients, but aliasing occurs only for terms of lower order, between $(0,0)$ and $(6,6)$. Some terms are integrated correctly because of parity or because $(6,6)$ is sufficiently close to the approximation basis for the terms to satisfy the criterion of Theorem 4.11, $\Psi_{\mathbf{j}'}, \Psi_{\mathbf{j}} \in \mathcal{A}(\mathcal{Q}_{\mathbf{k}}^{(d)})$ for some $\mathbf{k} \in \mathcal{K}$.*

Direct quadrature makes the same mistakes, but is not limited to errors below $(6,6)$ because Theorem 5.5 has different conditions; hence the aliasing is more severe.

This example is representative of the general case. Direct quadrature always incurs at least as much external aliasing as the Smolyak approach, and the methods become equivalent if the external term causing aliasing is of very high order. Although both methods exhibit external aliasing onto coefficients of the approximating basis (the “internal coefficients”), in a realistic problem (with suitable decay and truncation) the magnitude of the external coefficients and the corresponding corruption should be much smaller than the internal coefficients themselves. Hence, the error introduced in the approximation is manageable.

FIGURE 5.3. Numerical results for Example 5.6; each color square indicates the log of the coefficient magnitude for the basis function at its position. The circle indicates the correct non-zero coefficient. The Smolyak pseudospectral approach has fewer terms corrupted by external aliasing in this case.



5.4. Summary of comparison. Compared to the Smolyak pseudospectral approach, direct quadrature yields larger internal *and* external aliasing errors. Because of these aliasing errors, direct quadrature is essentially unable to make efficient use of most sparse quadrature rules. The Smolyak pseudospectral approach, on the other hand, is guaranteed never to have internal aliasing if the one-dimensional pseudospectral operators are chosen according to simple guidelines. We therefore recommend against using direct quadrature.

6. Adaptive Polynomial Expansions. When constructing a polynomial approximation of a black-box computational model, there are two essential questions: first, which basis terms should be included in the expansion; and second, what are the coefficients of those basis terms? The Smolyak construction allows detailed control over the truncation of the polynomial expansion and the work required to compute it. Since we typically do not have *a priori* information about the functional relationship generated by a black-box model, we develop an adaptive approach to tailor the Smolyak approximation to this function, following the dimension-adaptive quadrature approach of Gerstner & Griebel [13].

The Smolyak algorithm is well suited to an adaptive approach. The telescoping sum converges as the index set grows, so we can simply add more terms to improve the approximation until we are satisfied. We separate our adaptive approach into two components: local improvements to the Smolyak multi-index set and a global stopping criteria.

6.1. Local adaptivity. Local adaptivity is responsible for identifying multi-indices to add to a Smolyak multi-index set in order to improve its accuracy. A standard approach is simply to grow an isotropic simplex of side length n . Gerstner & Griebel instead suggest a series of greedy refinements that customize the Smolyak algorithm to a particular problem [13].

The local refinement used in [13] is to select a multi-index $\mathbf{k} \in \mathcal{K}$ and to add the forward neighbors of \mathbf{k} that are admissible. The multi-index \mathbf{k} is selected via an error indicator $\epsilon(\mathbf{k})$. We follow [13] and assume that whenever \mathbf{k} has a large impact on the result of the algorithm, it represents a “direction” that likely needs further refinement.

Let \mathbf{k} be a multi-index such that $\mathcal{K}' := \mathcal{K} \cup \mathbf{k}$, where \mathcal{K} and \mathcal{K}' are admissible multi-index sets. The triangle inequality (for some appropriate norm) bounds the change in the Smolyak approximation produced by adding \mathbf{k} to \mathcal{K} , yielding a useful error indicator:

$$\|A(\mathcal{K}', d, \mathcal{L}) - A(\mathcal{K}, d, \mathcal{L})\| \leq \|\Delta_{k_1}^1 \otimes \cdots \otimes \Delta_{k_d}^d\| =: \epsilon(\mathbf{k}) \quad (6.1)$$

Conveniently, this error indicator does not change as \mathcal{K} evolves, so we need only compute it once. At each adaptation step, we find the \mathbf{k} that maximizes $\epsilon(\mathbf{k})$ and that has at least one admissible forward neighbor. Then we add those forward neighbors.

6.2. Global adaptivity. Now that we have a strategy to locally improve a multi-index set, it is useful to have a global estimate of the error of the approximation, ϵ_g . We cannot expect to compute the exact error, but even an indicator which is correlated with the error is useful. We follow Gerstner & Griebel's choice of global error indicator

$$\epsilon_g := \sum \epsilon(\mathbf{k}) \quad (6.2)$$

where the sum is taken over all the multi-indices that are eligible for local adaptation at any particular step (i.e., that have admissible forward neighbors).

6.3. Comments. Thus far we have presented the adaptation strategy without reference to the problem of polynomial approximation. In this specific context, we use the $L^2(\mathbf{X}, w)$ norm in (6.1) because it corresponds to the convergence properties of pseudospectral approximation.

We conclude this section with a few observations about the adaptive approach. First, it is a heuristic method and we can only recommend it based on the success of adaptive sparse quadrature and experimental validation (see Section 9). Second, the greedy approach assumes that following the largest local errors leads to the largest global sources of error. Our analysis of external aliasing suggests that in the case of pseudospectral approximation, significant missing terms alias onto some of the *included* lower-order coefficients, giving the algorithm a useful indication of the direction to refine. Finally, Gerstner & Griebel suggest augmenting the local refinement algorithm to add terms that are computationally inexpensive, even if they do not appear immediately useful. In general, balancing the error indicator (6.1) with an indicator of the computational cost associated with a multi-index \mathbf{k} might improve efficiency for certain problems, or avoid pathological cases where ϵ is otherwise locally zero, but we do not explore this extension here.

7. Selection of Quadrature Rules. Thus far we have sidestepped practical questions about which quadrature rules exist or are most efficient. Our analysis has relied only on polynomial accuracy of quadrature rules; all quadrature rules with a given polynomial accuracy allow the same truncation of a pseudospectral approximation. In practice, however, we care about the cumulative cost of the adaptive algorithm, which must step through successive levels of refinement.

Integration over a bounded interval with uniform weighting offers the widest variety of quadrature choices, and thus allows a thorough discussion. Table 7.1 summarizes the costs of several common quadrature schemes. First, we see that linear-growth Gaussian quadrature is asymptotically much less efficient than exponential-growth in reaching any particular accuracy. However, for rules with fewer than about ten points, this difference is not yet significant. Second, Clenshaw-Curtis shows efficiency equivalent to exponential-growth Gaussian: both use n points to reach n th order polynomial accuracy [5]. However, their performance with respect to external aliasing differs: Clenshaw-Curtis slowly loses accuracy if the integrand is of order greater than n , while Gaussian quadrature gives $\mathcal{O}(1)$ error even on $(n+1)$ -order functions [34]. This may make Clenshaw-Curtis Smolyak pseudospectral estimates more efficient. Finally, we consider Gauss-Patterson quadrature, which is nested and has significantly higher polynomial accuracy—for a given cumulative cost—than the other types [26]. Computing the quadrature points and weights in finite precision (even extended-precision) arithmetic has practically limited Gauss-Patterson rules to 255 points, but we recommend them whenever this is sufficient.

For most other weights and intervals, there are fewer choices that provide precise polynomial accuracy, so exponential-growth Gaussian quadrature is our default choice. In the specific case of Gaussian weight, Genz has provided a family of Kronrod extensions, similar to Gauss-Patterson quadrature, which may be a useful option [12].

REMARK 7.1. *If a linear growth rule is chosen and the domain is symmetric, we suggest that each new level include at least two points, so that the corresponding basis grows by at least one even and one odd basis function. This removes the possibility for unexpected effects on the adaptive strategy if the target function is actually even or odd.*

Order	Lin. G				Exp. G				C-C			G-P	
	p	a	t		p	a	t		p	a		p	a
1	1	1	1		1	1	1		1	1		1	1
2	2	3	3		2	3	3		3	3		3	5
3	3	5	6		4	7	7		5	5		7	10
4	4	7	10		8	15	15		9	9		15	22
5	5	9	15		16	31	31		17	17		31	46
6	6	11	21		32	63	63		31	31		63	94
m	m	$2m-1$	$m^2-m/2$		2^{m-1}	2^m-1	2^m-1		$2^{m-1}+1$	$2^{m-1}+1$			

TABLE 7.1

The cost of four quadrature strategies as their order increases: linear growth Gauss-Legendre quadrature (Lin. G), exponential growth Gauss-Legendre quadrature (Exp. G), Clenshaw-Curtis quadrature (C-C), and Gauss-Patterson quadrature (G-P). We list the number of points used to compute the given rule (p), the polynomial accuracy (a), and the total number of points used so far (t). For nested rule, (p) = (t), so the total column is omitted.

8. Connections to Interpolation. It is well known that in one dimension, pseudospectral approximation based on Gaussian quadrature is equivalent to Lagrange interpolation on the same quadrature points [4, 16, 3]. This connection to interpolation is used to prove the convergence of pseudospectral operators. Similarly, the tensor product of pseudospectral approximations based on Gaussian quadrature is equivalent to Lagrange interpolation on the full tensor grid. Unfortunately, the Smolyak pseudospectral algorithm we have described does not actually produce an interpolant, because none of the quadrature rules is both nested (Clenshaw-Curtis, Gauss-Patterson) *and* yields an interpolant (Gaussian quadrature).

There is an extension to Lagrange interpolation called *sparse interpolation*, which uses information from a sparse grid and has strong known convergence properties [23, 2]. Constantine *et al.* [6] show that the Smolyak pseudospectral method using Gaussian quadrature is equivalent to sparse interpolation. It is important to note that sparse interpolants are not necessarily interpolants at every point on the sparse grid; instead, they are Smolyak combinations of full tensor interpolants. If the constituent full tensor rules are interpolants on nested point sets, then the Smolyak algorithm produces an interpolant on the corresponding sparse grid [29].

The connection of Gaussian quadrature-based pseudospectral operators to interpolation is useful for analysis, but it is an open question whether these rules are actually more efficient in practice, as compared to other options—specifically Clenshaw-Curtis or Gauss-Patterson.

9. Numerical Experiments. Our numerical experiments focus on evaluating the performance of different quadrature rules and of the adaptive Smolyak approximation strategy. Aside from the numerical examples of Section 5, we do not investigate the performance of direct quadrature any further. Given our theoretical analysis of aliasing errors and the numerical demonstrations of Constantine *et al.*, one can conclude without further demonstration that destructive internal aliasing indeed appears in practice.

We begin by evaluating mean-square convergence on the Genz test functions. Then we approximate a chemical kinetic system, to illustrate the efficiency and accuracy of adaptive Smolyak pseudospectral methods.

9.1. Basic convergence: Genz functions. The Genz family comprises six parameterized functions, defined on $[-1, 1]^d \rightarrow \mathbb{R}$ [10, 11]. They are commonly used to investigate the accuracy of quadrature rules and interpolation schemes [2, 19]. The purpose of this example is to show that different Smolyak pseudospectral strategies behave roughly as expected, as evidenced by decreasing L^2 approximation errors as more function evaluations are employed.

Our first test uses five isotropic and non-adaptive pseudospectral approximation strategies. The first strategy is the isotropic full tensor pseudospectral algorithm, based on Gauss-Legendre quadrature, where the order grows exponentially with level. The other four strategies are total-order expansions of increasing order based on the following quadrature rules: linear growth Gauss-Legendre, exponential growth Gauss-Legendre, Clenshaw-Curtis, and Gauss-Patterson. All the rules were selected so the final rule would have around 10^7 points.

We consider 20 random realizations of each Genz function in $d = 5$ dimensions; random parameters for the Genz functions were generated as in [19]. Each estimate of L^2 approximation error is computed by Monte Carlo sampling with 10^3 points. Figure 9.1 plots L^2 error at each stage, where each point represents the mean error over the 20 random functions.

We can draw relatively simple conclusions from this data. All the methods show fast convergence for the first four *continuous* Genz functions, indicating that the internal aliasing issues have indeed been resolved. In contrast, one would expect direct quadrature to suffer from large aliasing errors for the three super-linear growth rules. Otherwise, judging the efficiency of the different rules is not prudent, because differences in truncation and the functions themselves obscure differences in efficiency. In deference to our adaptive strategy, we ultimately do not recommend this style of isotropic and function-independent truncation anyway. We do note, however, that full tensor rules never perform better than the others; this is to be expected, as the Genz functions are not completely coupled. The last two functions are not continuous, so all the methods converge much more slowly, again as expected.

To test our adaptive approach, Figure 9.2 shows results from a similar experiment, now comparing the convergence of an adaptive Smolyak pseudospectral algorithm with that of a non-adaptive algorithm. For consistency, both are based on Gauss-Patterson quadrature. As we cannot synchronize the number of evaluations used by the adaptive algorithm for different functions, we plot individual errors for the 20 random functions instead of the mean error. This reveals the variability in difficulty of the functions, which was hidden in the previous plot. We conclude that the adaptive algorithm also converges as expected, with performance comparable to the non-adaptive algorithm. We do not expect much improvement from adaptivity here, because the Genz functions are essentially isotropic with smoothly decaying coefficients, so there is little structure for the adaptive algorithm to exploit. Although omitted here for brevity, other quadrature rules produce similar results.

9.2. Adaptivity: chemical kinetics. To illustrate the benefits of an adaptive Smolyak approach, we build a surrogate for a realistic simulation of a combustion kinetics problem. Specifically, we consider the auto-ignition of a methane-air mixture given 14 uncertain rate parameters. Governing equations for this process are a set of stiff nonlinear ODEs expressing conservation of energy and of chemical species [18]. The uncertain rate parameters represent activation energies of reactions governing the conversion of methane to methyl, each endowed with a uniform distribution varying over $[0.8, 1.25]$ of the nominal value. These parameters appear in Arrhenius expressions for the species production rates, with the reaction pathways and their nominal rate parameters given by the GRIMech 3.0 mechanism [15]. The output of interest is the logarithm of the ignition time, which is a functional of the trajectory of the ODE system. Simulations were performed with the help of the TChem software library [27], which provides convenient evaluations of thermodynamic properties and species production rates, along with Jacobians for implicit time integration.

Chemical kinetics are an excellent testbed for adaptive approximation because, by the nature of detailed kinetic systems, we expect strong coupling between some inputs and weak coupling between others, but we cannot predict these couplings *a priori*. We test the effectiveness of adaptive Smolyak pseudospectral methods based on the four quadrature rules discussed earlier. As our earlier analysis suggested that Gauss-Patterson quadrature should be most efficient, our basis of comparison is a non-adaptive Gauss-Patterson total-order Smolyak pseudospectral expansion. We ran the non-adaptive algorithm with a total order index set truncated at $n = 5$ (which includes monomial basis terms up through $\psi_{23}^{(i)}$), using around 40000 point evaluations and taking over an hour to run. We tuned the four adaptive algorithms to terminate with approximately the same number of evaluations.

Figure 9.3 compares convergence of the five algorithms. The L^2 errors reported on the vertical axis are Monte Carlo estimates using 1000 points. Except for a small deviation at fewer than 20 model evaluations, all of the adaptive methods significantly outperform the non-adaptive method. The performance of the different quadrature rules is essentially as predicted in Section

7: Gauss-Patterson is the most efficient, exponential growth Gauss-Legendre and Clenshaw-Curtis are nearly equivalent, and linear growth Gauss-Legendre performs worse as the order of the polynomial approximation increases. Compared to the non-adaptive algorithm, adaptive Gauss-Patterson yields more than two orders of magnitude reduction in the error at the same number of model evaluations. Linear growth Gaussian quadrature is initially comparable to exponential growth Gaussian quadrature, because the asymptotic benefits of exponential growth do not appear while the algorithm is principally using very small one-dimensional quadrature rules. At the end of these experiments, a reasonable number of higher order quadrature rules are used and the difference becomes visible.

We conclude by illustrating that the adaptive algorithm is effective because it successfully focuses its efforts on high-magnitude coefficients—that is, coefficients that make the most significant contributions to the function. Even though the non-adaptive expansion has around 37,000 terms and the final adaptive Gauss-Patterson expansion only has about 28,000 terms, the adaptive expansion exhibits much lower error because most of the additional terms in the non-adaptive expansion are nearly zero. By skipping many near-zero coefficients, the adaptive approach is able to locate and estimate a number of higher-order terms with large magnitudes. Figure 9.4 depicts this pattern by plotting the difference between the numbers of included terms in the final adaptive Gauss-Patterson and non-adaptive expansions. The adaptive algorithm does not actually add any higher order monomials; neither uses one-dimensional basis terms of order higher than $\psi_{23}^{(i)}$. Instead, the adaptive algorithm adds mixed terms of higher total order, thus capturing the coupling of certain variables in more detail than the non-adaptive algorithm. The figure shows that terms through 30th order are included in the adaptive expansion, all of which are products of non-constant polynomials in more than one dimension.

10. Conclusions. This paper gives a rigorous development of Smolyak pseudospectral algorithms, a practical approach for constructing polynomial chaos expansions from point evaluations of a function. A common alternative approach, direct quadrature, has previously been shown to suffer from large errors. We explain these errors as a consequence of internal aliasing and delineate the exact circumstances, derived from properties of the chosen polynomial basis and quadrature rules, under which internal aliasing will occur. Internal aliasing is a problem inherent to direct quadrature approaches, which use a single (sparse) quadrature rule to compute a set of spectral coefficients. These approaches fail because they substitute a numerical approximation for only a portion of the algorithm, i.e., the evaluation of integrals, without considering the impact of this approximation on the entire construction. For almost all sparse quadrature rules, internal aliasing errors may be overcome only through an inefficient use of function evaluations. In contrast, the Smolyak pseudospectral algorithm computes spectral coefficients by assembling tensor-product pseudospectral approximations in a coherent fashion that avoids internal aliasing by construction; moreover, it has smaller external aliasing errors. To establish these properties, we prove that the accurate set of a Smolyak pseudospectral approximation contains a union of the accurate sets of all its constituent tensor-product approximation operators. These results are applicable to any choice of quadrature rule and generalized sparse grid, and are verified through numerical demonstrations.

A key strength of Smolyak algorithms is that they are highly customizable through the choice of admissible multi-index sets. To this end, we describe a simple alteration to the adaptive sparse quadrature approach of Gerstner & Griebel [13], creating a corresponding method for adaptive pseudospectral approximation. Numerical experiments then evaluate the performance of different quadrature rules and of adaptive versus non-adaptive pseudospectral approximation. Tests of the adaptive method on a realistic chemical kinetics problem show multiple order-of-magnitude gains in accuracy over a non-adaptive approach.

While the adaptive approach illustrated here is deliberately simple, many extensions are possible. For instance, as described in Section 6.3, measures of computational cost may be added to the local refinement criterion. One could also use the gradient of the L^2 error indicator to identify optimal directions in the space of multi-indices along which to continue refinement, or

to avoid adding all the forward neighbors of the multi-index selected for refinement. These and other developments will be pursued in future work.

A flexible open-source C++ code implementing the adaptive approximation method discussed in this paper is available at <https://bitbucket.org/mituq/muq/>.

Acknowledgments. The authors would like to thank Paul Constantine for helpful discussions and for sharing a preprint of his original paper, which inspired this work. We would also like to thank Omar Knio and Justin Winokur for many helpful discussions, and Tom Coles for help with the chemical kinetics example. P. Conrad was supported during this work by a Department of Defense NDSEG Fellowship and an NSF graduate fellowship. P. Conrad and Y. Marzouk acknowledge additional support from the Department of Energy, Office of Advanced Scientific Computing Research (ASCR).

REFERENCES

- [1] I. BABUSKA, F. NOBILE, AND R. TEMPONE, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1005–1034.
- [2] VOLKER BARTHELMANN, ERICH NOVAK, AND KLAUS RITTER, *High dimensional polynomial interpolation on sparse grids*, Advances in Computational Mathematics, 12 (2000), pp. 273–288.
- [3] JOHN P. BOYD, *Chebyshev and Fourier Spectral Methods*, Dover Publications, 2nd revised ed., 2001.
- [4] CLAUDIO CANUTO, M. YOUSUFF HUSSAINI, ALFIO QUARTERONI, AND THOMAS A. ZHANG JR., *Spectral Methods: Fundamentals in single domains*, Springer Berlin Heidelberg, 2006.
- [5] C.W. CLENSHAW AND A.R. CURTIS, *A method for numerical integration on an automatic computer*, Numerische Mathematik, 2 (1960), pp. 197–205.
- [6] PAUL G. CONSTANTINE, MICHAEL S. ELDRED, AND ERIC T. PHIPPS, *Sparse Pseudospectral Approximation Method*, Computer Methods in Applied Mechanics and Engineering, 229–232 (2012), pp. 1–12.
- [7] M. S. ELDRED AND J. BURKARDT, *Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification*, AIAA paper 2009–0976, (2009).
- [8] OLIVER G. ERNST, ANTJE MUGLER, HANS-JÖRG STARKLOFF, AND ELISABETH ULLMANN, *On the convergence of generalized polynomial chaos expansions*, ESAIM: Mathematical Modeling and Numerical Analysis, 46 (2012), pp. 317–339.
- [9] W. GANDER, *Change of basis in polynomial interpolation*, Numerical Linear Algebra with Applications, 12 (2005), pp. 769–778.
- [10] ALAN GENZ, *Testing Multidimensional Integration Routines*, in Tools, Methods, and Languages for Scientific and Engineering Computation, B. Ford, J.C. Rault, and F. Thomasset, eds., North-Holland, 1984, pp. 81–94.
- [11] ———, *A Package for Testing Multiple Integration Subroutines*, in Numerical Integration: Recent Developments, Software and Applications, Patrick Keast and Graeme Fairweather, eds., D Reidel, 1987, pp. 337–340.
- [12] ———, *Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight*, Journal of Computational and Applied Mathematics, 71 (1996), pp. 299–309.
- [13] T. GERSTNER AND M. GRIEBEL, *Dimension-Adaptive Tensor-Product Quadrature*, Computing, 71 (2003), pp. 65–87.
- [14] ROGER GHANEM AND POL SPANOS, *Stochastic finite elements: a spectral approach*, Springer-Verlag, 1991.
- [15] ZHIWEI QIN GREGORY P. SMITH, DAVID M. GOLDEN, MICHAEL FRENKLACH, NIGEL W. MORIARTY, BORIS EITENEER, MIKHAIL GOLDENBERG, C. THOMAS BOWMAN, RONALD K. HANSON, SOONHO SONG, WILLIAM C. GARDINER, JR., VITALI V. LISSIANSKI, *GRI-MECH 3.0*. http://www.me.berkeley.edu/gri_mech.
- [16] JAN S. HESTHAVEN, SIGAL GOTTLIEB, AND DAVID GOTTLIEB, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, 2007.
- [17] XUN HUAN AND YOUSSEF M. MARZOUK, *Simulation-based optimal Bayesian experimental design for nonlinear systems*, Journal of Computational Physics (in press), (2012).
- [18] ROBERT J. KEE, MICHAEL ELLIOTT COLTRIN, AND PETER GLARBORG, *Chemically Reacting Flow: Theory and Practice*, Wiley-Interscience, 2003.
- [19] ANDREAS KLIMKE AND BARBARA WOHLMUTH, *Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 561–579.
- [20] O. P. LE MAITRE AND OMAR M. KNIO, *Spectral Methods for Uncertainty Quantification*, Springer, 2010.
- [21] A. MARCH AND K. WILLCOX, *Constrained multifidelity optimization using model calibration*, Structural and Multidisciplinary Optimization, 46 (2012), pp. 93–109.
- [22] YOUSSEF M. MARZOUK, HABIB N. NAJM, AND LARRY A. RAHN, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, Journal of Computational Physics, 224 (2007), pp. 560–586.
- [23] F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *A Sparse Grid Stochastic Collocation Method for Partial*

- Differential Equations with Random Input Data*, SIAM Journal on Numerical Analysis, 46 (2007), p. 2309.
- [24] ERICH NOVAK AND KLAUS RITTER, *High dimensional integration of smooth functions over cubes*, Numerische Mathematik, 75 (1996), pp. 79–97.
 - [25] ———, *Simple Cubature Formulas with High Polynomial Exactness*, Constructive Approximation, 15 (1999), pp. 499–522.
 - [26] T. N. L. PATTERSON, *The Optimum Addition of Points to Quadrature Formulae*, Mathematics of Computation, 22 (1968), p. 847.
 - [27] COSMIN SAFTA, HABIB NAJM, AND OMAR KNIO, *TChem*. <http://www.sandia.gov/tchem/>.
 - [28] WINFRIED SICKEL AND TINO ULLRICH, *The Smolyak algorithm, sampling on sparse grids and function spaces of dominating mixed smoothness*, East J Approx, 13 (2007), pp. 387–425.
 - [29] ———, *The Smolyak algorithm, sampling on sparse grids and function spaces of dominating mixed smoothness*, East J Approx, 13 (2007), pp. 387–425.
 - [30] ———, *Tensor products of Sobolev-Besov spaces and applications to approximation from the hyperbolic cross*, Journal of Approximation Theory, 161 (2009), pp. 748–786.
 - [31] S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. nauk SSSR, 4 (1963), pp. 240–243.
 - [32] C. SOIZE AND R. GHANEM, *Physical Systems with Random Uncertainties: Chaos representations with arbitrary probability measure*, SIAM Journal of Scientific Computing, 26 (2004), pp. 395–410.
 - [33] BRUNO SUDRET, *Global sensitivity analysis using polynomial chaos expansions*, Reliability Engineering & System Safety, 93 (2008), pp. 964–979.
 - [34] LLOYD N. TREFETHEN, *Is Gauss Quadrature Better than Clenshaw-Curtis?*, SIAM Review, 50 (2008), p. 67.
 - [35] G. W. WASILKOWSKI AND H. WOZNIAKOWSKI, *Explicit cost bounds of algorithms for multivariate tensor product problems*, Journal of Complexity, 11 (1995), pp. 1–56.
 - [36] ———, *Weighted Tensor Product Algorithms for Linear Multivariate Problems*, Journal of Complexity, 15 (1999), pp. 402–447.
 - [37] ———, *Polynomial-Time Algorithms for Multivariate Linear Problems with Finite-Order Weights: Worst Case Setting*, Foundations of Computational Mathematics, 5 (2005), pp. 451–491.
 - [38] N. WIENER, *The Homogeneous Chaos*, American Journal of Mathematics, 60 (1938), pp. 897–936.
 - [39] DONGBIN XIU, *Efficient Collocational Approach for Parametric Uncertainty Analysis*, Communications in Computational Physics, 2 (2007), pp. 293–309.
 - [40] ———, *Fast numerical methods for stochastic computations: a review*, Communications in computational physics, 5 (2009), pp. 242–272.
 - [41] ———, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.
 - [42] DONGBIN XIU AND JAN S. HESTHAVEN, *High-Order Collocation Methods for Differential Equations with Random Inputs*, SIAM Journal on Scientific Computing, 27 (2005), p. 1118.
 - [43] DONGBIN XIU AND GEORGE EM KARNIAKAKIS, *The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM Journal on Scientific Computing, 24 (2002), p. 619.

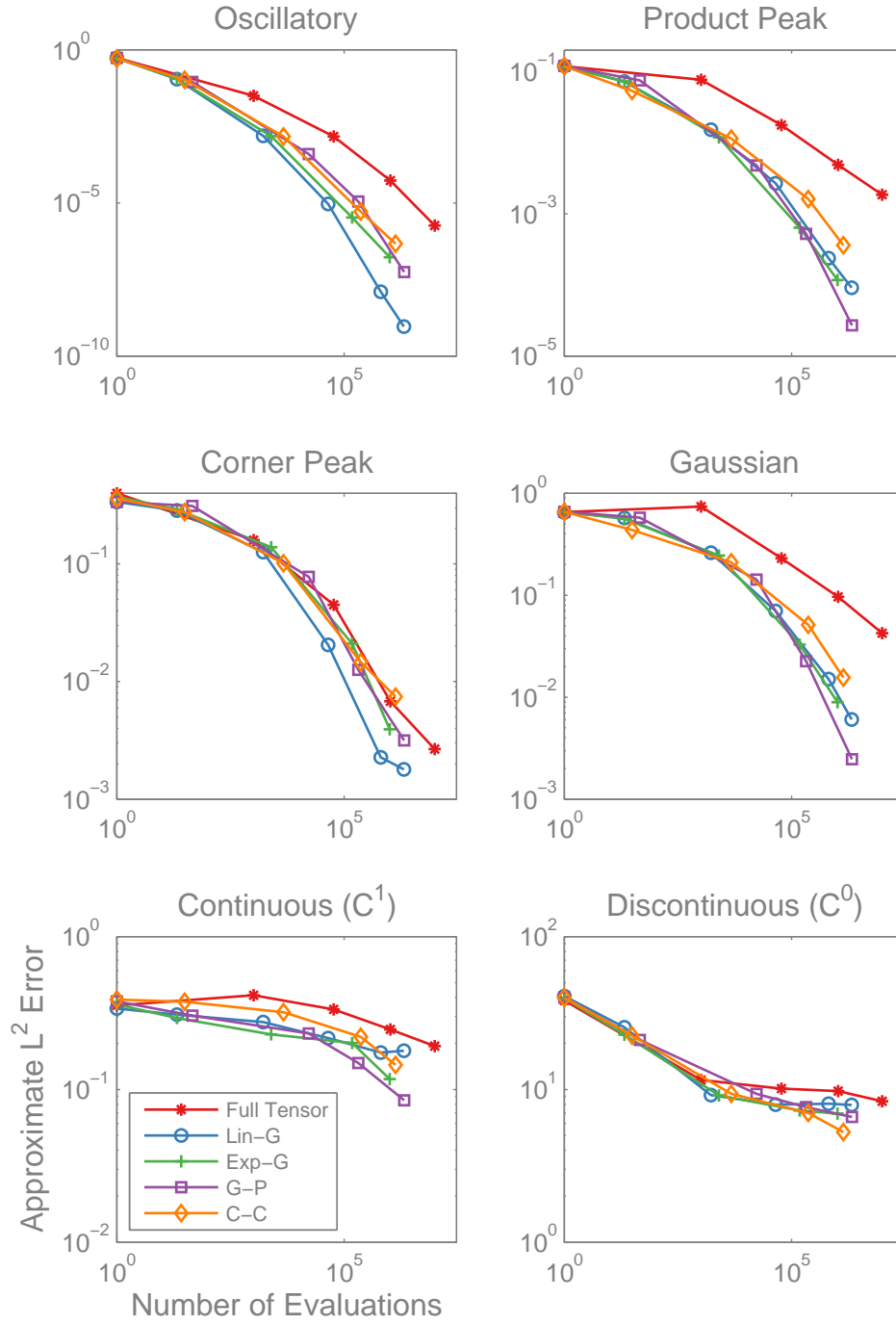


FIGURE 9.1. Mean L^2 convergence of the non-adaptive isotropic total-order Smolyak pseudospectral algorithm with various quadrature rules, compared to the full tensor pseudospectral algorithm, on the Genz test functions.

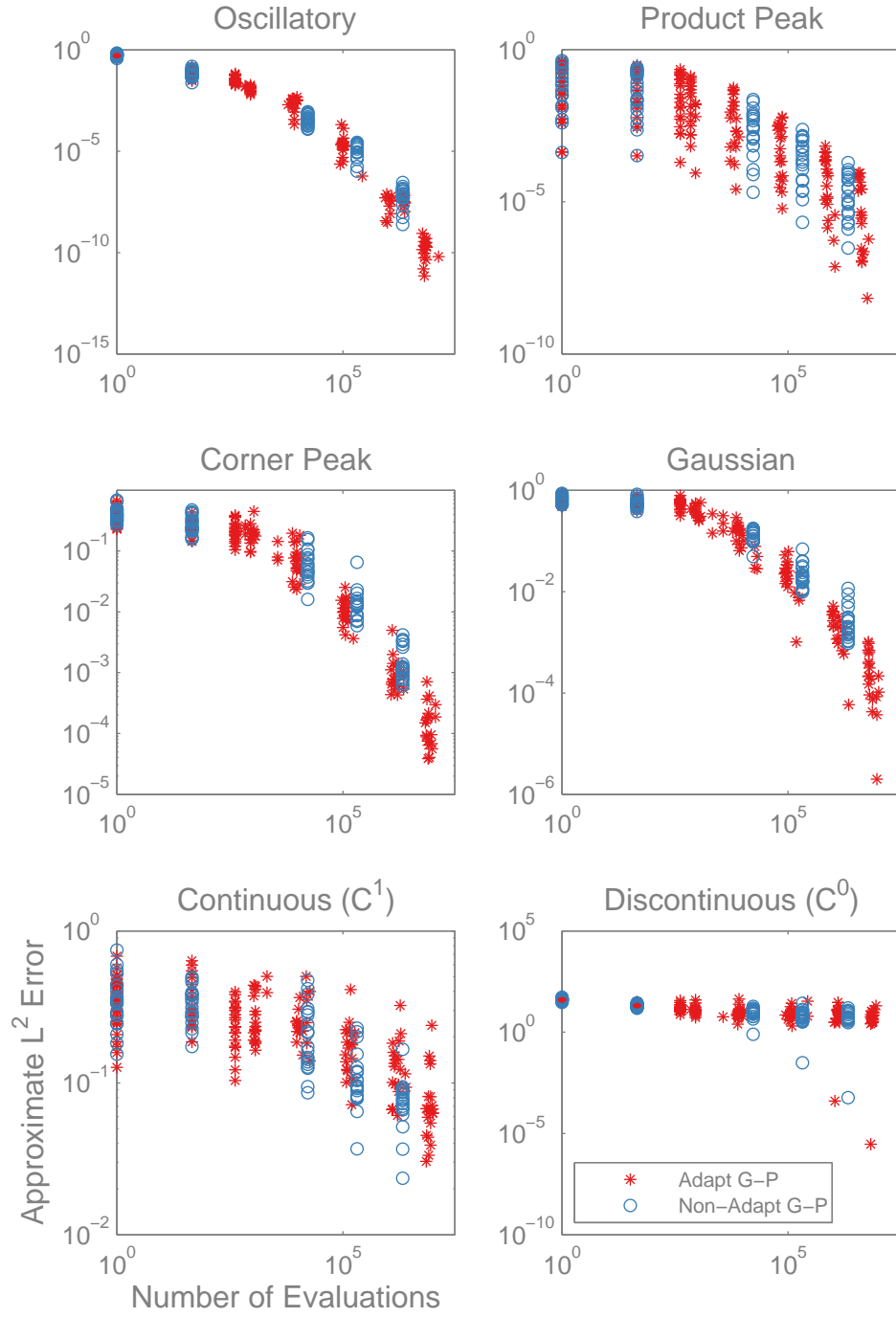


FIGURE 9.2. L^2 convergence of the adaptive and non-adaptive Gauss-Patterson Smolyak pseudospectral algorithm. Individual results for 20 random instances of the Genz functions are shown.

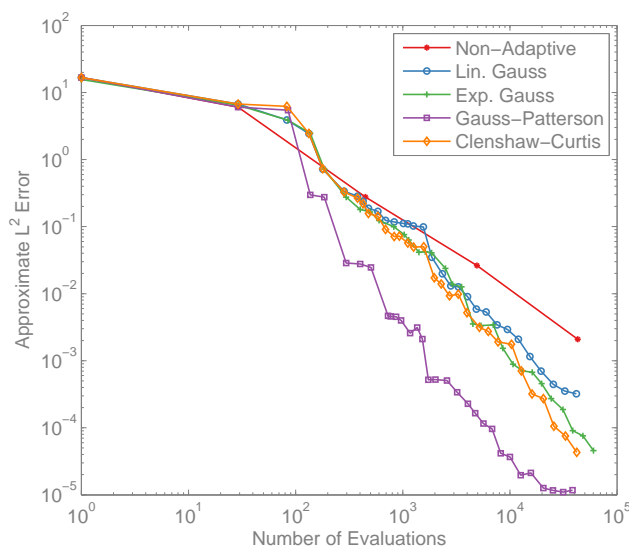


FIGURE 9.3. L^2 convergence of ignition delay in a 14-dimensional chemical kinetic system; comparing a {non-adaptive isotropic total-order Gauss-Patterson-based Smolyak pseudospectral algorithm} to the adaptive algorithm with various quadrature rules.

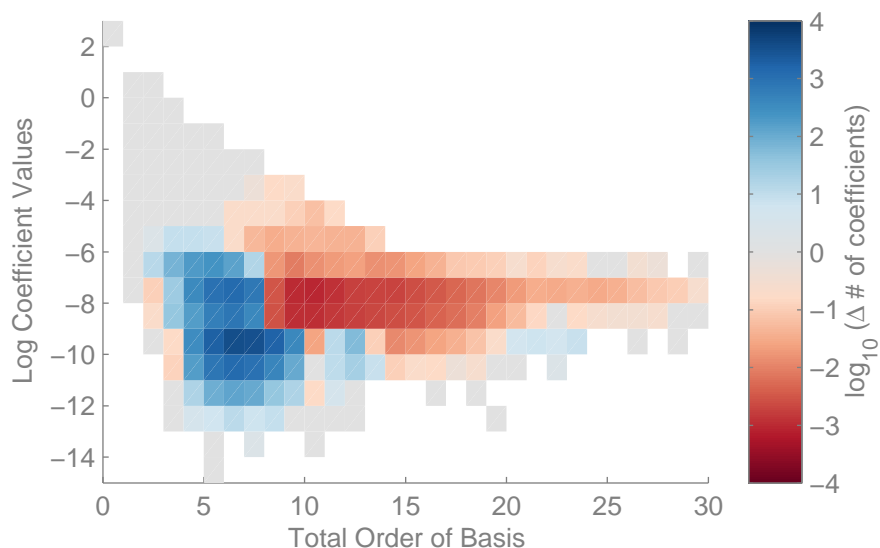


FIGURE 9.4. The plot depicts the difference between the number of coefficients of a particular magnitude and order in the final adaptive and non-adaptive Gauss-Patterson based expansions. The horizontal axis is the order of the term and the vertical axis specifies \log_{10} of the coefficient value. The color represents \log_{10} of the difference between the two methods, where positive values indicate more terms in the non-adaptive expansion. Hence, the dark blue at (6, -10) indicates that the non-adaptive expansion includes around 3,000 extra terms of magnitude 10^{-10} and the dark red at (10, -8) indicates that the adaptive expansion includes about 1,000 extra terms of magnitude 10^{-8} . Grey squares are the same for both expansions and white squares are not present in either.